

## Resum

En aquest treball de fi de grau es dissenya i desenvolupa una aplicació per a ordinador destinada a donar suport als logopedes en la rehabilitació logopèdica d'infants que tenen trastorns lingüístics, generalment causats per danys cerebrals que afecten a la regió del llenguatge. L'aplicació implementa una de les tasques que es duu a terme avui en dia manualment a l'hospital i que consisteix en la segmentació sil·làbica. A través de l'aplicació els nens i nenes entrenaran les seves habilitats lingüístiques tant semàntiques com de la parla ja que inclou un sistema de reconeixement de veu.

En primer lloc, es defineixen els objectius del projecte, es delimita l'abast que aquest tindrà, es fa un estudi de mercat de les aplicacions que existeixen actualment que presentin semblances amb la que es dissenya. En segon lloc, es fa un anàlisi dels usuaris per a determinar els requisits de l'aplicació a partir de les necessitats del usuari. I, en tercer lloc, es defineix la tasca i les propietats que haurà de contenir l'aplicació.

A continuació, es fa una introducció de les eines que s'empraran per a desenvolupar l'aplicació. Aquestes són la plataforma Unity amb el seu corresponent Motor de Videojocs, que servirà per implementar la tasca en llenguatge de programació C#, i un fitxer XML que actua de base de dades i conté tota la informació necessària per al seu correcte funcionament.

Seguidament, es fa un estudi dels tipus d'aplicacions de reconeixement de veu que existeixen al mercat per tal de triar quin s'ajustaria més a les necessitats d'aquesta aplicació i, en la part més important del joc, es fa una descripció de l'estructura interna de l'aplicació, el seu funcionament i característiques més importants.

L'aplicació que finalment s'obté és una part d'un projecte que englobarà varies tasques de les que es realitzen avui en dia manualment. Aquesta aplicació és la primera que es fa incorporant reconeixement de veu i servirà de precedent per a futures implementacions de noves tasques.

# Sumari

<b>RESUM</b>	<b>1</b>
<b>SUMARI</b>	<b>2</b>
<b>1. GLOSSARI</b>	<b>5</b>
<b>2. PREFACI</b>	<b>7</b>
2.1. Origen del projecte .....	7
2.2. Motivació .....	7
2.3. Requeriments previs .....	8
<b>3. INTRODUCCIÓ</b>	<b>9</b>
3.1. Objectius del projecte .....	9
3.2. Abast del projecte .....	10
<b>4. ESTUDI DE MERCAT</b>	<b>11</b>
4.1. <i>Iniciación a la lectoescriptura</i> [8] .....	11
4.1.1. Jocs .....	12
4.1.2. Avantatges i inconvenients .....	13
4.2. <i>Vocaliza</i> [10] .....	13
4.2.1. Jocs .....	14
4.2.2. Avantatges i inconvenients .....	15
4.3. <i>Speechviewer</i> [9] .....	16
4.3.1. Jocs .....	16
4.3.2. Avantatges i inconvenients .....	16
<b>5. ANÀLISI DELS USUARIS</b>	<b>18</b>
5.1. Infants amb trastorns lingüístics .....	18
5.2. Logopeda .....	19
<b>6. TASCA</b>	<b>20</b>
6.1. Què fa .....	20
6.2. Funcionament .....	21
6.2.1. Parametrització .....	22
6.2.2. Dinamisme .....	22
6.2.3. Visualització de resultats .....	23
6.2.4. Base de dades .....	23
6.2.4.1. Introducció de dades .....	23

<b>7. EINES</b>	<b>26</b>
7.1. Motor de joc - plataforma Unity	26
7.1.1. Unity	27
7.1.2. Entorn integrat de desenvolupament	30
7.1.2.1. Scripts	30
7.1.2.2. Classe MonoBehaviour	31
7.2. Processament de la informació	32
7.2.1. Fitxer XML	32
7.2.2. Fitxers TXT	35
<b>8. BASE TEÒRICA DEL RECONeixEMENT DE VEU</b>	<b>36</b>
8.1.1. Sistema de reconeixement de veu de Google	38
8.1.2. Vivo	39
8.1.3. Dynamic time warping (DTW)	41
8.1.4. Unity System Speech Recognition [13]	42
<b>9. ESTRUCTURA DEL JOC</b>	<b>43</b>
9.1. Pantalla_principal	43
9.1.1. Funcionament	44
9.1.2. GameObjects	45
9.1.2.1. Main Camera (càmera principal)	45
9.1.2.2. Fons	45
9.1.2.3. Títol	46
9.1.2.4. Jugar, opcions, resultats i sortir	46
9.2. Pantalla_opcions	48
9.2.1. Funcionament	49
9.2.2. GameObjects	50
9.2.2.1. Main Camera (càmera principal)	50
9.2.2.2. Fons	50
9.2.2.3. Opcions, nivell i idioma	50
9.2.2.4. Botons Baix, Mig, Alt, Català i Castellà	50
9.2.2.5. Botó torna	52
9.2.2.6. ObjecteBuit	53
9.3. Pantalla_joc	54
9.3.1. Funcionament	54
9.3.2. GameObjects	55

9.3.2.1. Fons .....	55
9.3.2.2. Main Camera (càmera principal) .....	55
9.3.2.3. Pregunta i nom .....	57
9.3.2.4. Números (1,2,3,4,5) .....	57
9.3.2.5. Canvas-imatge .....	59
9.3.2.6. Objectebuit1 .....	59
9.3.2.7. Altaveu i Següent .....	60
9.3.2.8. Objectebuit2 .....	60
9.3.2.9. Imatge .....	62
9.3.2.10. Canvas-tick i creu .....	63
9.3.2.11. Correcte/Incorrecte .....	63
9.3.2.12. Micro .....	63
9.4. Pantalla_resultats .....	65
9.4.1. Funcionament .....	66
9.4.2. GameObjects .....	66
9.4.2.1. Main Camera (càmera principal) .....	66
9.4.2.2. Fons .....	66
9.4.2.3. Resultats, Puntuació, Fallades i Encerts .....	66
9.4.2.4. Valor_puntuacio, Valor_encerts i Valor_fallades .....	67
9.4.2.5. Botó torna .....	67
<b>10. TREBALL FUTUR .....</b>	<b>69</b>
<b>11. PLANIFICACIÓ TEMPORAL .....</b>	<b>70</b>
<b>12. COSTOS .....</b>	<b>73</b>
<b>13. IMPACTE EN L'ENTORN .....</b>	<b>75</b>
<b>CONCLUSIONS .....</b>	<b>76</b>
<b>AGRAÏMENTS .....</b>	<b>77</b>
<b>BIBLIOGRAFIA .....</b>	<b>78</b>
Referències bibliogràfiques .....	78
Bibliografia complementària .....	79

# 1. Glossari

TFG: Treball de final de grau.

Unity: Motor de videojoc multi-plataforma creat per Unity Technologies.

UPC: Universitat Politècnica de Catalunya.

ETSEIB: Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.

Afàsia: Trastorn del llenguatge que es caracteritza per la incapacitat o dificultat de comunicar-se mitjançant la parla, l'escriptura i la mímica i que es deu a lesions cerebrals.

Disfàsia: Trastorn en l'adquisició del llenguatge que es manifesta en una alteració en l'expressió i comprensió del llenguatge.

API: Interfície de Programació d'Aplicacions. Conjunt de procediments, funcions i mètodes oferts per una determinada biblioteca per ser utilitzats per un altre software. IDE: Entorn de desenvolupament integrat.

Asset: Element de Unity que s'introdueix a un joc: materials, textures, personatges, scripts...

Classe: Construcció que permet crear tipus personalitzats propis mitjançant l'agrupació de variables d'altres tipus, mètodes i esdeveniments.

Codi: El codi d'un programa informàtic és un conjunt de línies de text llegibles per l'home que són les instruccions que ha de seguir la màquina per executar-lo.

Collider: Component de Unity que defineix la forma d'un objecte per als propòsits de col·lisions físiques.

CSharp (o C#): Llenguatge de programació orientat a objectes desenvolupat i estandarditzat per Microsoft com a part de la seva plataforma .NET i que després va ser aprovat com un estàndard per la ECMA i ISO.

GameObject: Objectes de Unity que podem introduir en la pantalla: cubs, esferes... i als quals podem incloure components que modificaran les seves característiques.

GUI: Interfície gràfica de l'usuari. Conjunt d'elements gràfics tal com finestres, botons i menús, que permeten la interacció entre l'usuari i l'aplicació informàtica.

Software: Part lògica i intangible que fa que la computadora pugui realitzar tasques i

controlar dispositius físics o hardware que la componen. Està format pels programes que fan que la computadora realitzi diferents tasques i funcions.

Hardware: Conjunt d'elements físics o materials que constitueixen un ordinador o un sistema informàtic.

IDE: Entorn de desenvolupament integrat. Programa informàtic compost per un conjunt d'eines de programació. És un entorn de programació que consta d'un editor de codi, un compilador, un depurador i un constructor d'interfície gràfica (GUI).

Motor de videojocs: Conjunt de rutines de programació enfocades al disseny, creació i representació d'un videojoc.

Script: Llista d'instruccions o ordres, escrites utilitzant un llenguatge informàtic.

Sistema operatiu: Conjunt de programes que duen a terme, segura i eficaçment, la gestió de recursos i que permeten que hi hagi una interfície entre el hardware i la persona que l'utilitza.

Struct: Tipus de valor que normalment es fa servir en programació per encapsular petits grups de variables relacionades, com les característiques d'un element en el joc que es desenvolupa en aquest projecte.

## 2. Prefaci

### 2.1. Origen del projecte

Avui en dia, s'està impulsant molt l'ús de recursos informàtics com a mètode d'aprenentatge per a nens i nenes, en el que s'anomenen jocs seriosos. Aquests tenen finalitats terapèutiques i pedagògiques entre d'altres i s'apliquen en àmbits molt diversos. En el cas d'infants els jocs seriosos fomenten l'aprenentatge actiu, el treball en equip i proporcionen un ambient lliure on poden desenvolupar les seves habilitats.

Alguns membres de *l'Hospital de Sant Joan de Déu de Barcelona* es van adonar de la necessitat de desenvolupar una aplicació per a infants que tenen problemes lingüístics, ja que actualment no existeix cap joc per a la rehabilitació logopèdica en català que pugui ser atractiu per a nens i nenes. Per aquest motiu es van posar en contacte amb el Grup d'Informàtica per a l'Enginyeria (GIE) de la UPC per a proposar-los un projecte que permetés analitzar la viabilitat de construcció d'un joc que compleixi amb els requeriments que es demanen.

Actualment, el tractament en aquest tipus de pacients es fa a partir d'una maleta de jocs que conté un conjunt de tasques que realitza l'infant juntament amb el logopeda. Aquest treball de fi de grau consisteix en la implementació informàtica d'una d'aquestes tasques. L'ús de l'aplicació està destinat a pacients de l'hospital que pateixen patologies lingüístiques.

Amb aquest projecte es pretén que tant el pacient com el terapeuta disposin d'una eina de suport, que es pugui adaptar al coneixement de l'usuari i que permeti portar un control de la seva evolució. A més, l'eina permet la pràctica autònoma del pacients sense la necessitat que l'especialista estigui present durant tota la sessió.

### 2.2. Motivació

La realització del present treball de fi de grau ha estat motivat per diversos motius:

- L'interès per l'àmbit de l'enginyeria biomèdica. Aquest és un sector que està guanyant molta importància i que està avançant molt en els últims temps ja que es busca el benestar i l'autonomia dels éssers humans. El treball pretén ser un primer contacte amb aquest àmbit i servir com a precedent per a projectes futurs.
- Formar part d'un projecte conjunt, elaborat per diferents persones, amb una finalitat enriquidora com és la d'ajudar a nens i nenes que tinguin dificultats lingüístiques i

contribuir en certa mesura en la seva millora.

- El repte que suposa treballar amb un entorn de programació del que no se'n tenien coneixements previs, així com el desenvolupament des de zero d'una aplicació. Aquest fet permet posar en pràctica els coneixements teòrics que s'han obtingut al llarg del grau.
- Per últim, el fet de resoldre un problema real per tu mateix sense que tingui un objectiu únicament acadèmic incrementa l'entusiasme i les ganes d'intentar arribar el més lluny possible.

## 2.3. Requeriments previs

El treball requereix coneixements bàsics d'informàtica i de programació informàtica per a la seva correcta realització. En ell s'apliquen coneixements que s'han adquirit al llarg del Grau d'Enginyeria en Tecnologies Industrials cursat a l'ETSEIB, com ara fonaments d'informàtica, informàtica, organització i gestió, gestió de projectes i tecnologia del medi ambient.

A més a més, gran part de la informació que es pot trobar a la xarxa es troben en anglès i, per tant, es requereixen uns mínims coneixements d'aquest idioma per a la seva comprensió.

Com tot treball que es duu a terme avui en dia, aquest treball requereix un ús solent de la informació per tal d'extreure els coneixements útils de la xarxa i alhora trobar les fonts més fiables per a l'autoaprenentatge.



## 3. Introducció

### 3.1. Objectius del projecte

L'objectiu principal del projecte és crear una aplicació apta per a ordinador que contingui jocs o taques i que serveixi com a eina de suport per als terapeutes a l'hora de tractar nens i nenes de l'hospital en la seva rehabilitació logopèdica. Es pretén que el joc abasti diversos aspectes de la rehabilitació en les diferents tasques, conformant tot el conjunt un joc ampli en el que hi hagi nivells i es en el que el logopeda pugui triar la tasca o joc que ha de practicar el nen.

A continuació s'introdueixen els objectius funcionals de l'aplicació, que ha de cobrir les necessitats tant de pacients com de logopedes.

- Oferir una eina que estigui dirigida a nens en un rang d'edats de 4 a 16 anys i que sigui atractiva per a ells.
- L'aplicació ha de ser flexible per tal que els logopedes puguin renovar i ampliar els continguts del joc (gràfics, sons, etc). Per aquest motiu ha de contenir una base de dades, accessible pel logopeda, que contingui tota la informació referent al joc.
- Tot el contingut del joc ha d'estar disponible en dos idiomes, català i castellà per tal que els usuaris treballin en la llengua amb la que se sentin més còmodes.
- Les tasques que s'han d'implementar han de provenir de la maleta de jocs o tasques proporcionada pels especialistes de *l'Hospital Sant Joan de Déu*, que corresponen a les que es desenvolupen avui presencialment i de forma conjunta amb el terapeuta. Aquestes tasques han estat àmpliament provades i se n'ha pogut garantir la validesa i efectivitat.
- L'aplicació ha de ser adaptable als coneixements de cada infant. Per aquest motiu s'ha d'incorporar paràmetres variables per adaptar el joc als nens i nenes.
- S'han de mostrar els resultats obtinguts per cada usuari quan aquest finalitza la tasca en qüestió per tal que el logopeda pugui portar un control de la millora del pacient.
- Ha de ser accessible per a tots els usuaris i facilitar el treball a distància i el control remot.

- Algunes de les tasques han de permetre la participació oral de l'usuari, tot i que és més important la teràpia semàntica que la fonètica.
- El funcionament del joc ha de ser senzill i intuïtiu.

La tasca específica que es vol implementar al llarg d'aquest treball de fi de grau inclou per primera vegada una part de reconeixement de veu, per a que els nens i nenes puguin interaccionar mitjançant la veu amb el joc i d'aquesta manera desenvolupin la comunicació oral. El treball exercirà la funció de prova pilot per tal de verificar si els recursos informàtics amb els que es dissenya l'aplicació disposen d'eines suficients per a l'implementació d'aquesta tasca i, per tant, aquest serà també un dels objectius d'aquest projecte.

### **3.2. Abast del projecte**

Aquest treball de fi de grau consisteix en el desenvolupament d'una de les tasques que contindrà el joc final. Té una estructura similar a la dels treballs de fi de grau realitzats anteriorment per el Carles Regué i l'Albert Sauri, que van implementar altres tasques de rehabilitació logopèdica que s'inclouran en el joc complet, tot i que en aquest TFG el que s'analitza és diferent del que s'havia fet en els dos casos anteriors.

El projecte abasta les fases de disseny, programació i implementació de l'aplicació tot i que, degut al breu període de realització del projecte, 4 mesos, i a la manca d'espai adient on fer-ho, aquest no inclou la fase de validació amb pacients reals. Aquesta fase es durà a terme quan el projecte estigui finalitzat en el seu conjunt, inclogui totes les tasques i estigui llest per al seu ús.

En aquest projecte s'ha volgut construir un prototip de tasca completa que inclogui tots els requisits comentats a l'apartat 3.1 i que incorpori reconeixement de veu. S'ha decidit implementar-la només per a ordinadors, deixant la possibilitat de fer-la hàbil per a mòbils el un futur.

## 4. Estudi de mercat

Abans de fer un projecte per cobrir una necessitat, cal estudiar quines opcions existeixen al mercat avui en dia per tal d'estudiar i determinar per quina causa no han estat ben acceptades o quina mancança presenten per a cobrir les necessitats dels usuaris. Per aquest motiu s'ha fet un estudi de les aplicacions que existeixen al mercat actualment que poden tenir una finalitat similar a la del projecte que es vol dur a terme i s'ha estudiat tant el funcionament com l'adequació d'aquestes al públic a qui va dirigit aquest projecte.

Es pot trobar una gran varietat d'aplicacions similars a la xarxa malgrat cap d'elles compleix tots els requisits que ha de complir l'aplicació que s'ha de dissenyar (ap. 3.1). Entre elles destaquen tres aplicacions que o bé són les més conegudes i usades en l'àmbit de la logopèdia o bé són aplicacions, amb prestacions similars a les desitjades, que s'ha considerat oportunes de destacar.

Seguidament, es fa un recull d'aquestes tres aplicacions tot donant informació dels avantatges i inconvenients que poden presentar i una valoració de si podrien adaptar-se de manera adequada al present treball. Aquestes aplicacions són *Iniciación a la lectoescriptura*, el projecte *Vocaliza* y l'aplicació *SpeechViewer*.

### 4.1. *Iniciación a la lectoescriptura* [8]

L'aplicació de *Iniciación a la lectoescriptura* es troba a la web de la Junta d'Andalusia, concretament a l'àrea de la conselleria d'educació. Està desenvolupada i dissenyada pel director i programador Francisco López Hijano i pel dissenyador gràfic Francisco Vega Sepúlveda, dos autònoms espanyols que han col·laborat en el desenvolupament d'aplicacions, per a diverses comunitats autònomes, que fomenten l'aprenentatge de la lectura i l'escriptura.

L'aplicació conté activitats interactives d'atenció, percepció visual, lectura, escriptura i comprensió i està destinada a alumnes d'educació infantil i primària, però també a infants amb Necessitats Educatives Especials que necessiten realitzar activitats que estimulin i madurin les àrees que intervenen en l'adquisició de la lectoescriptura. Les seves característiques principals són:

- Idioma: Castellà
- Objectiu: estimulació i maduració de les àrees que intervenen en l'adquisició de la lectoescriptura

- Gran varietat d'activitats
- Disponibilitat d'activitats per a la pràctica de diferents àmbits.
- Aplicació utilitzable des de qualsevol sistema operatiu que tingui instal·lat el reproductor Macromedia Flash Player

#### 4.1.1. Jocs

L'aplicació disposa d'una àmplia gamma de jocs essent alguns més adequats que d'altres per a la rehabilitació logopèdica. El conjunt d'activitats està format per activitats interactives d'atenció, percepció visual, lectura, escriptura i comprensió.

Hi ha un total de 19 activitats diferents, entre les quals n'hi ha alguna que consta de diverses tasques en el seu interior, i que poden ser tant variades com per exemple trobar l'intrús en un grup d'imatges o emparellar imatges de temàtica similar. Es recullen tots en el següent llistat:

- |                    |                      |                  |                      |
|--------------------|----------------------|------------------|----------------------|
| • Siluetes         | • Absurds            | • Pintar         | • Trencaclosques     |
| • Formes           | • Emparella          | • Llegir         | • Escriure           |
| • Retrat robot     | • Ordenar per mida   | • Busca l'intrús | • Circuits de vocals |
| • Ordenar història | • Buscar diferències | • Buscar iguals  | • Buscar síl·labes   |
| • Música           | • Conte 1            | • Conte 2        |                      |



L'aplicació no té inclòs cap joc que tingui reconeixement de veu ni participi l'usuari de manera oral.

#### 4.1.2. Avantatges i inconvenients

En aquest apartat es fa un recull dels possibles avantatges i inconvenients que presentaria aquesta aplicació per a l'ús específic que se li vol donar i de l'adequació d'aquesta per als usuaris que tindrà.

Avantatges	Inconvenients
Eina atractiva per a nens i nenes	No està orientada específicament a nens amb problemes lingüístics
Gràfics i jocs molt variats	El funcionament de les tasques és diferent en cada cas i pot dificultar la comprensió
Gran interacció amb l'usuari	No té reconeixement de veu
Sons clars	No té emmagatzematge de resultats
Vocabulari molt variat	No es pot adaptar el nivell al grau de coneixement dels usuaris
	Només en castellà
	No s'ha provat la validesa i efectivitat d'aquestes tasques, contràriament a les de la maleta de jocs de l'hospital

Tot i que l'eina és molt completa i atrau molt l'atenció dels infants, si es para atenció al llistat d'inconvenients es pot comprovar que l'aplicació no compleix amb alguns requisits bàsics com són la disponibilitat dels idiomes català i castellà, el control dels resultats per part del terapeuta i l'adaptació al grau de coneixement de l'usuari.

#### 4.2. Vocaliza [10]

Vocaliza és una aplicació desenvolupada per la *Universidad de Zaragoza*, emprada únicament en l'àmbit de la rehabilitació logopèdica. Aquesta aplicació està destinada a recolzar el tractament de rehabilitació dut a terme per part dels logopedes i facilitar-los una eina de suport.

El seu ús està orientat principalment cap a infants amb problemes lingüístics i només està disponible en llengua castellana. L'aplicació treballa tres nivells diferents de llenguatge, que són el fonològic, el semàntic i el sintàctic. Destaca principalment per d'incorporació d'un sistema fonètic, de reconeixement de veu, que permet la interacció oral de l'usuari i així aquest pot desenvolupar l'habilitat fonològica a més de la comprensió. Aquest tret és clarament diferenciador d'altres aplicacions ja que requereix una elaboració lenta i acurada fet que fa que no tots els jocs l'adoptin.

A continuació es fa un recull de les característiques principals que l'aplicació presenta:

- Idioma: Castellà
- Objectiu: donar suport al logopeda, facilitant-li l'ús de l'aplicació i un ventall prou gran de jocs per a entrenar les diferents àrees del llenguatge
- Incorporació de reconeixement de veu
- Disponibilitat de tasques per a l'estimulació de l'aprenentatge en més d'una branca de la comunicació
- Personalització del joc. Aquest guarda tota la informació de l'usuari en una carpeta que es pot traslladar d'un ordinador a un altre
- Model acústic adaptat a l'usuari que permet millorar el funcionament del sistema i conèixer l'avaluació i millora de l'usuari al llarg del temps quan el pacient pateix alguna patologia greu de la parla

#### **4.2.1. Jocs**

Com s'ha comentat anteriorment el tret distintiu d'aquesta aplicació és el de la incorporació de reconeixement de veu i és en aquest tret en el que es basen pràcticament totes les tasques que conté l'aplicació encara que també incorpora els nivells de semàntica i sintàctic en les seves activitats.

En total l'aplicació consta de 4 jocs amb funcionament diferent entre sí però clars i entenedors per als usuaris. Aquests 4 jocs consisteixen en el següent:

- Joc de pronunciació: té com a objectiu l'entrenament de la parla a nivell fonològic. Consisteix en la pronunciació correcta d'una paraula de manera que el programa estudia la proximitat entre la pronunciació de l'usuari i el model, obtenint com a resultat el grau d'aproximació.
- Joc de record: té com a objectiu l'entrenament de la parla a nivell fonològic. El joc

consisteix en que l'usuari pot pronunciar lliurement la paraula que vulgui entrenar sense haver-ne visualitzat cap imatge ni sentit cap àudio.

- Joc de la frase: té com a objectiu l'entrenament del nivell sintàctic i fonològic del llenguatge. Es tracta de pronunciar de forma ordenada les paraules que conté la frase donada pel joc.
- Joc de l'enigma: té com a objectiu l'entrenament del nivell semàntic i fonològic del llenguatge. Consisteix en la tria de la imatge que doni resposta a la pregunta que es planteja i, seguidament, pronunciar-la.

#### 4.2.2. Avantatges i inconvenients

En aquest apartat es fa un recull dels possibles avantatges i inconvenients que presentaria aquesta aplicació per a l'ús específic que se li vol donar i de l'adequació d'aquesta per als usuaris que tindrà.

Avantatges	Inconvenients
Eina atractiva per a nens i nenes	No té emmagatzematge de resultats
Incorpora reconeixement de veu fet que permet una interacció oral amb l'usuari	No es pot adaptar el nivell al grau de coneixement dels usuaris, cal modificar la base de dades
Model acústic adaptat a l'usuari	Només en castellà
Informació de l'usuari enregistrada en una carpeta. Cada usuari té vocabulari concret i per tant el joc és personalitzat	Poca varietat d'activitats
Vocabulari molt variat	Activitats poc variades i molt centrades només en la parla

Tenint en compte tots els aspectes mencionats, es pot arribar a la conclusió de que *Vocaliza* és una aplicació molt completa pel que fa al desenvolupament de la parla del pacient ja que ha estat dissenyada per nens sords i muts, malgrat és força incompleta pel que fa a altres àrees del llenguatge.

A més a més, aquesta aplicació està feta en una única llengua, el castellà, i per tant no compleix un dels requisits bàsics que havia de complir l'aplicació per tal de satisfer les necessitats de l'*Hospital Sant Joan de Déu* de Barcelona.

### 4.3. Speechviewer [9]

Speechviewer és una aplicació creada per IBM l'any 1989 i que consisteix en un visualitzador fonètic dissenyat amb l'objectiu d'afavorir el desenvolupament de l'expressió oral i va ser creat per a ajudar als professionals del llenguatge, la parla i la veu en la rehabilitació de problemes lingüístics.

El programa incorpora una sèrie d'activitats que es poden emprar en pacients que pateixen patologies relacionades amb l'audició, la parla o el llenguatge. Està caracteritzat per:

- Idioma: anglès
- Objectiu: proporcionar un millor domini i desenvolupament de l'expressió de la parla. Com a objectius secundaris té: fer possible la percepció de les qualitats de les paraules, educar la respiració i corregir la prosòdia a la veu
- Incorporació de reconeixement de veu
- Emmagatzematge de les dades i anàlisi estadístic, per a facilitar el control i seguiment de les activitats del pacient
- Paràmetres de la parla modificables en funció del pacient

#### 4.3.1. Jocs

L'aplicació disposa d'un conjunt de jocs que permeten desenvolupar l'expressió oral, tractant un aspecte específic en cadascuna d'elles. Aquestes activitats permeten emmagatzemar i analitzar la parla, la representació visual de la senyal de veu en el temps i la seva reproducció.

Els jocs sumen un total de dotze, en els que es treballa la presència de so, la gamma d'intensitat del so, la presència de veu, la duració de la veu i l'escala de veus, entre d'altres. Tots ells es poden trobar agrupats en funció del paràmetre de la parla al que fan referència: to, prosòdia, intensitat, sonoritat i fonologia. El programa representa els paràmetres mitjançant objectes que són atractius per l'usuari, com ara animals i joguines.

#### 4.3.2. Avantatges i inconvenients

En aquest apartat es fa un recull dels possibles avantatges i inconvenients que presentaria aquesta aplicació per a l'ús específic que se li vol donar i de l'adequació d'aquesta per als usuaris que tindrà.



Avantatges	Inconvenients
Eina atractiva per a nens i nenes	No es pot adaptar el nivell al grau de coneixement dels usuaris
Incorpora reconeixement de veu fet que permet una interacció oral amb l'usuari	No utilitza el català en el desenvolupament dels exercicis
Paràmetres acústics adaptats a l'usuari	Activitats molt centrades només en la parla
Emmagatzematge i anàlisi estadístic dels resultats	
Vocabulari molt variat	

De l'estudi d'aquesta aplicació es pot concloure que aquesta és una aplicació força completa, però succeeix el mateix que amb el *Vocaliza* ja que les activitats estan molt centrades en el desenvolupament de la parla mentre que en manquen que estiguin relacionades amb el nivell semàntic i sintàctic del llenguatge. Tampoc Speechviewer ha estat dissenyada per a nens amb danys cerebrals.

## 5. Anàlisi dels usuaris

Per a dissenyar qualsevol eina ja sigui física o informàtica cal determinar el públic al que va dirigida i els usuaris que interaccionaran amb el dispositiu de manera que aquest pugui adaptar-se al màxim a les necessitats dels usuaris i sigui el més segur, útil i còmode possible. Per aquest motiu, en aquest apartat es procedeix a fer una classificació dels usuaris de l'aplicació que es dissenyarà i els que hi estaran en contacte.

Com s'ha comentat a l'apartat 2.1, l'origen d'aquest projecte es troba en la necessitat de donar solució a un buit en el mercat pel que fa a aplicacions destinades a la rehabilitació logopèdica per a infants. Per aquest motiu, es considera que l'usuari raó de ser de l'aplicació són els infants de l'hospital amb trastorns lingüístics ja que aquesta els ha de servir per a desenvolupar les seves habilitats i d'aprenentatge.

També s'ha considerat usuari raó de ser de l'aplicació als logopedes perquè l'aplicació els servirà de suport i perquè interaccionaran contínuament amb ella, supervisant la feina que faci el pacient, renovant la base de dades, ampliant-la i fent un control i seguiment.

A continuació es detallarà les característiques i necessitats dels esmentats usuaris raó de ser.

### 5.1. Infants amb trastorns lingüístics

Existeixen molts tipus de trastorns lingüístics que poden fer a un infant susceptible de necessitar el tractament d'un especialista, és a dir, un logopeda i, per tant, de poder necessitar l'ús de l'aplicació. Alguns d'aquests problemes poden ser:

- Trastorn específic del llenguatge (TEL) o disfasia: alteració en el desenvolupament del llenguatge tenint un desenvolupament normal en la resta de paràmetres evolutius
- Afàsia: trastorn caracteritzat per la dificultat de comunicar-se mitjançant la parla, l'escriptura o la mímica i que és deguda a lesions cerebrals
- Mutisme selectiu: negació a parlar en determinades situacions
- Anormalitats en la forma de parlar com ara el tartamudeig, la disartia, la dislalia i trastorns de la prosodia, que ocasionen una parla entretallada i no fluida
- Trastorns per paràlisi cerebral
- Trastorns fonològics

- Trastorns de memòria i atenció
- Trastorns d'aprenentatge de lectura com la dislèxia

L'aplicació que es desenvoluparà està destinada concretament a infants amb trastorns d'afàsia. Aquest es dona majoritàriament per infeccions del sistema nerviós, accidents vasculars i traumatismes craneoencefàlics que provoquen danys cerebrals, afectant a les capacitats comunicatives de l'infant.

Les dificultats lingüístiques que pateixen les persones amb aquest trastorn poden ser expressives i receptives, dificultat en l'ús de paraules i oracions i dificultat a l'hora de comprendre als altres, respectivament, o bé ambdues. L'infant presenta reduïda rapidesa mental i dificultat per assimilar conceptes, per la qual cosa és molt important que el joc tingui un funcionament senzill, que les imatges i els textos es visualitzin correctament i que el nivell es pugui adaptar als coneixements de cada usuari.

Si els danys cerebrals són lleus la persona sovint pot recuperar les seves habilitats lingüístiques, en altres casos no les pot recuperar completament però sí que les pot treballar i desenvolupar per tal de rehabilitar el màxim de coneixements possibles. El tractament el duu a terme el logopeda, que s'explicarà seguidament.

## 5.2. Logopeda

El logopeda és el pedagog de la parla i el llenguatge que s'encarrega del tractament dels trastorns lingüístics i juga un paper molt important en el funcionament de l'aplicació. Aquest és el que s'encarrega de la supervisió i control dels pacients, no només adaptant-los el joc al nivell de cada usuari sinó també verificant els resultats que obtenen i les habilitats que van adquirint.

L'aplicació pretén ser una eina de suport per a aquests professionals, que els permeti estalviar temps a l'hora de tractar cada pacient i serveixi com a eina addicional per a la millora de les seves habilitats.

A més a més, el logopeda és l'encarregat de modificar la base de dades en cas de que ho requereixi, aportant així una bateria de recursos més ampla i renovada. Per aquest motiu cal que l'aplicació consti d'aclariments per als logopedes sobre com emprar la base de dades i, a més, el seu ús sigui simple per a poder comprendre el funcionament general de l'aplicació.

## 6. Tasca

En aquest apartat s'explica en detall en què consisteix la tasca que es vol implementar, com s'ha arribat a ella i quin serà el seu funcionament general.

La tasca que es desenvolupa en aquest projecte és una de les que es trobava dins de la maleta de tasques proporcionades per l'Hospital Sant Joan de Déu, que corresponen a les que duen a terme actualment de manera manual i presencial. La idea i definició de la tasca està generada doncs per professionals en la matèria i, per tant, es garanteix que l'aplicació resultant exerceixi la funció que es desitja sobre el pacient i que en cap cas pugui ser perjudicial.

### 6.1. Què fa

La tasca consisteix en la segmentació sil·làbica d'un grup de paraules donat. El grup de paraules està format per un conjunt aleatori que prové d'una base de dades en la que hi ha emmagatzemada tota la informació de l'aplicació. Per cada paraula es dona una imatge relacionada amb la paraula, un àudio amb la pronunciació de la paraula, i un text, en què hi ha la paraula escrita.

El pacient haurà de determinar el nombre de síl·labes que conté la paraula, pronunciar el més clarament possible el número i seleccionar el número mitjançant els botons. A partir dels valors obtinguts a l'àudio es determinarà si el número pronunciat s'aproxima al correcte i a partir del botó que premin es determinarà si el número de síl·labes és el que li correspon a la paraula.

A més, l'aplicació porta un sistema de puntuació que estimula els pacients per intentar-se superar a ells mateixos. El sistema de puntuació adjudica 5 punts per paraula encertada i en resta 1 per paraula fallada.

Amb aquesta activitat no només es vol entrenar la parla sinó que també es volen desenvolupar habilitats semàntiques. L'exercici de la separació sil·làbica fa que el pacient profunditzi en els sons de les paraules i amplii els seus coneixements del llenguatge.

Cal destacar que en un principi es volia unificar el funcionament del reconeixement de veu amb el de selecció del número de síl·labes de manera que el programa fos capaç d'interpretar el número pronunciat i determinés si era el correcte o no. Finalment, això no ha estat possible ja que el mètode de reconeixement de veu ha donat forces problemes i s'ha cregut convenient separar les tasques.

## 6.2. Funcionament

S'ha procurat que l'aplicació compleixi amb tots els requisits que s'havien imposat i que satisfaci les necessitats dels usuaris, per això s'ha dotat l'aplicació amb un sistema de parametrització, s'ha fet que l'aplicació sigui dinàmica i permeti una interacció amb l'usuari per a que li cridi l'atenció, un sistema de visualització dels resultats per a que l'especialista pugui seguir el progrés del pacient i una base de dades per actualitzar el contingut de l'aplicació.

El primer sistema, el de parametrització, consisteix en incorporar al sistema uns paràmetres variables en funció de l'usuari per tal que l'aplicació s'adapti a l'usuari tant pel que fa a l'idioma com al grau de coneixement i habilitat del pacient.

El segon, el dinamisme, permet que l'activitat sigui entretinguda i no esdevingui monòtona amb el pas del temps. Els infants són actius i inquiets i per aquest motiu cal una eina que els cridi l'atenció i que permeti la seva interacció contínua.

Finalment, el tercer sistema, el de visualització dels resultats consisteix en la visualització de la puntuació que s'ha obtingut al final del joc, així com el nombre d'encerts i errors i el temps que ha transcorregut entre l'inici i el final de l'aplicació. Així el logopeda pot portar un seguiment de la millora i el desenvolupament del pacient.

A continuació s'inclou una taula que mostra breument el resum del que s'ha comentat en aquest apartat:

Què fa?	Explicació	Paràmetres	Estructura
Segmentar en síl·labes una paraula donada	<p>El jugador llegeix, visualitza i sent, en cas que ho desitgi, una paraula.</p> <p>Seguidament, ha de dir (oralment) en quantes síl·labes es pot segmentar la paraula.</p> <p>Quan troba el nombre correcte: gratificació</p>	<p>- Possibilitat de triar l'idioma de funcionament de la tasca</p> <p>- Dificultat de les paraules, en funció de la seva llargada.</p>	Encert amb la pronunciació correcte del número de síl·labes.

### 6.2.1. Parametrització

Com es pot observar a la taula resum anterior, l'aplicació disposa de dos paràmetres que són modificables i adaptables a l'usuari. Encara que el funcionament és el mateix per a totes les opcions i paràmetres, el fet de poder adaptar l'aplicació a l'usuari fa que aquest connecti més i se senti més estimulat a jugar-hi.

La llengua de funcionament de l'aplicació és un dels paràmetres regulables ja que convé que el pacient desenvolupi la seva llengua materna o la que usa més habitualment, que és amb la que se sent més còmode i més important per a la seva comunicació amb la resta de persones. Com a Catalunya hi ha dos idiomes oficials, el català i el castellà, és lògic que l'aplicació pugui disposar de les dues llengües per al desenvolupament de l'aplicació.

L'altre paràmetre que es pot regular és el nivell, que està subdividit en tres grups: nivell baix, mig i alt. El nivell baix està conformat per paraules d'una o dues síl·labes, el nivell mig per paraules d'una, dues o tres síl·labes i el nivell alt per paraules d'una, dues, tres, quatre i cinc síl·labes. Aquest fet permet adaptar el nivell de dificultat de l'aplicació al jugador impedit doncs que el pugui sobrepassar i desmoralitzar-lo o bé que li resulti massa senzilla i poc interessant.

L'encarregat de la determinació d'aquests paràmetres és el logopeda, que pot modificar les opcions del joc mitjançant els botons de la pantalla d'opcions.

Idioma		Nivell		
Català	Castellà	Baix	Mig	Alt

Les opcions que té l'aplicació per defecte són l'idioma català i el nivell mig, per tant, en cas de que el logopeda consideri més adients altres opcions haurà d'accedir a elles i modificar-les

### 6.2.2. Dinamisme

Tot joc té una característica molt important, que és el dinamisme d'aquest. Cal que un joc sigui entretingut i àgil per tal que el jugador no es cansi i el joc no esdevingui monòton. Per aconseguir que l'aplicació d'aquest treball sigui el més dinàmica possible se l'ha dotat de diverses característiques: l'aleatorietat de les paraules que apareixen i l'existència d'animacions diverses.

Creant una base de dades àmplia i variada s'aconsegueix que el joc no resulti monòton ja que l'aleatorietat de la visualització de les imatges genera la sensació de que cada vegada que s'inicialitza el joc, aquest és diferent. Aquest fet fa que els pacients no s'apreguin les

solucions de memòria, que ampliïn el seu vocabulari i practiquin amb un vocabulari ric i divers.

L'aplicació també conté animacions i estímuls a la pantalla del joc, que fan que el jugador rebi una gratificació en fer-ho correctament i una penalització en fer-ho malament, tot i que sempre es prioritzen els estímuls positius per sobre dels negatius.

### **6.2.3. Visualització de resultats**

Un dels requisits d'aquesta aplicació era dotar-la d'un sistema de visualització de resultats per facilitar el seguiment de la millora del pacient. Com s'ha comentat a l'ap. 6.2.2, per a estimular el pacient s'ha introduït en el joc un sistema de puntuació i un recompte dels encerts i fallades, els resultats dels quals es podran observar a la pantalla de resultats.

S'accedeix a aquesta pantalla automàticament un cop finalitzat el joc i és accessible des de la pantalla principal en qualsevol moment, sempre que no es reiniciï el joc. La informació extreta de la pantalla de resultats és significativa a l'hora d'estudiar l'efectivitat que està tenint l'aplicació en l'aprenentatge del pacient i es pot emprar inclús per a fer un anàlisi estadístic i una seguiment temporal per veure l'evolució del jugador.

### **6.2.4. Base de dades**

L'especialista té ple control sobre el contingut del joc, podent-lo modificar i adaptar a cada usuari mitjançant una base de dades. Aquesta base de dades conté tota la informació relacionada amb el joc: el nom de les paraules que es visualitzaran amb els seus corresponents àudios, imatges, nombre de síl·labes i idioma.

S'ha considerat oportú que l'especialista pugui modificar i afegir el contingut d'aquesta base de dades en funció de cada usuari, per a entrenar amb el vocabulari que més pugui necessitar el pacient o bé per evitar el que es comentava a l'ap. 6.2.2 sobre la monotonia i modificar periòdicament el contingut de l'aplicació.

El sistema de modificació de les dades s'ha intentat que sigui el més senzill i còmode possible per facilitar la tasca del logopeda, tenint en compte que aquest pot no tenir coneixements d'informàtica.

#### **6.2.4.1. Introducció de dades**

Com s'ha comentat anteriorment és el logopeda el que té la tasca d'omplir la base de dades del joc per tal de poder controlar i personalitzar el contingut de l'aplicació. El mètode d'introducció és senzill i consta de diversos passos. Per facilitar la feina del logopeda s'ha realitzat el següent manual d'instruccions:

1. La tasca es basa en l'aparició d'imatges amb el nom de l'objecte que es mostra, per tant, en primer lloc caldrà fer una tria del vocabulari que el logopeda consideri que el pacient ha de practicar. Per a cada paraula el logopeda crearà un bloc amb 5 característiques que la definiran:

Característica	1	2	3	4	5
Paraula 1	Nom	Imatge	So	Síl·labes	Nivell

2. Ubicar totes les imatges del vocabulari que es vol tractar a la carpeta Assets i anomenar-les pel nom d'imatge amb el que després es cridarà aquesta.
3. Crear un arxiu d'àudio (mp3) en el que se senti la paraula que es visualitza. El nom de l'arxiu cal que sigui el nom de la imatge afegint-hi a darrera la paraula "So". Per exemple si es tractés de la paraula aigua, l'àudio s'hauria d'anomenar aiguaSo.
4. Obrir el fitxer XML que està ubicat a la carpeta Assets del projecte. Es pot obrir amb el Bloc de Notes, però és molt recomanable instal·lar-se el programa XML Notepad perquè facilita la comprensió del contingut de l'arxiu.



Fig. 6.2. Esquema de l'estructura interna del fitxer XML

5. El logopeda s'ha de posicionar a dins de la carpeta de dadesCAT o dadesCAST en funció de l'idioma de les dades que vulgui introduir. Per afegir informació a les carpetes cal que creï tants elements fills com paraules vulgui afegir ja que a cada paraula li correspondrà un element. Per afegir elements fills cal que es posicioni sobre la carpeta dadesCAT o dadesCAST i que premi la pestanya Insert del menú (o el botó dret del ratolí), que seleccioni l'opció element i child. El nom d'aquests elements no és rellevant i es pot triar amb total llibertat.
6. Introduir 5 atributs als elements creats. Aquests atributs es corresponen amb les característiques de cadascuna de les paraules que havia de definir en el primer pas el logopeda.

- Nom: conté la paraula que descriu la imatge.



- Foto: conté el nom de la imatge que s'ha guardat a la carpeta Assets.
- So: conté el nom de l'àudio que s'ha guardat a la carpeta Assets.
- Síll·labes: conté un número que és igual al nombre de síll·labes de la imatge.
- Nivell: conté un número indicatiu del grau de dificultat de la paraula. El criteri que es seguirà per determinar el nivell s'explicarà més endavant.

Per exemple, a la següent imatge es mostra l'estructura que tindrà un element (a mà esquerra) i el seu contingut (a mà dreta).

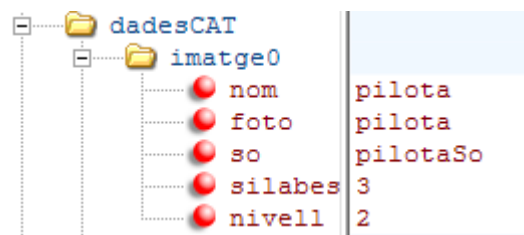


Fig. 6.3. Esquema de l'estructura interna d'un element

7. Per últim s'ha de guardar el fitxer després d'haver-lo modificat.

Aquest arxiu es pot modificar tantes vegades com el logopeda desitgi. De fet, és recomanable que ho faci regularment per a que el joc no esdevingui repetitiu.

Actualment la base de dades conté 24 i 23 elements en les carpetes dadesCAT i dadesCAST respectivament però aquestes no són fixes, és a dir, es poden eliminar i modificar segons el criteri del logopeda.

## 7. Eines

Per al desenvolupament de l'aplicació s'han utilitzat varies eines informàtiques, que són:

- Plataforma Unity, per a la implementació de l'aplicació, la seva visualització i la determinació del seu comportament
- Fitxer XML, per a la base de dades
- Fitxers TXT, emprats per emmagatzemar la llista de freqüències que s'obtenen dels àudios per a fer el reconeixement de veu

Cal remarcar que l'aplicació requereix de l'existència d'altaveus i de micròfon en l'ordinador per tal que es pugui dur a terme el correcte enregistrament de les freqüències dels àudios en els fitxers de text.

A continuació hi ha l'explicació detallada del funcionament del motor de joc i de la plataforma Unity, la seva metodologia i estructura interna, així com l'explicació del mètode de funcionament del fitxer XML que conforma la base de dades i els fitxers TXT d'enregistrament de freqüències.

### 7.1. Motor de joc - plataforma Unity

En aquest apartat s'exposa en què consisteix un motor de joc i el mètode de funcionament de la plataforma Unity, quina és la seva estructura i per quins elements i components pot estar formada.

Un motor de videojoc és un software que està dissenyat per crear i desenvolupar videojocs tant per a ordinadors com per dispositius mòbils. Tots els motors han d'oferir una funcionalitat bàsica que consisteix en proporcionar un motor de renderitzat ("render") en 2D i 3D, un motor que detecti la col·lisió física d'objectes i la resposta a la col·lisió, sons, música, animacions, comunicació amb la xarxa i gestió de la memòria.

Les capacitats gràfiques del motor són una de les claus per a la seva elecció. En aquest cas, al ser una aplicació que formarà part d'un projecte ampli, cal que totes les aplicacions estiguin desenvolupades amb la mateixa plataforma o motor de joc. Es va considerar que la plataforma més adient seria la plataforma Unity ja que disposa de molts recursos gràfics i és una plataforma que s'està emprant cada vegada més per aquest tipus de tasques i que dona bons resultats.

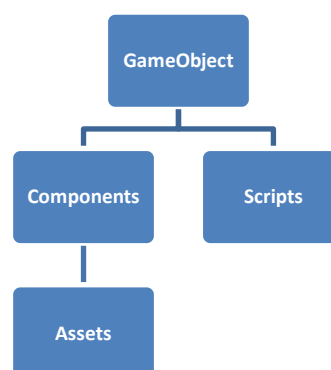
### 7.1.1. Unity

Abans d'explicar com s'ha desenvolupat l'aplicació, cal tenir uns coneixements bàsics sobre el funcionament de la plataforma de Unity, amb la que es desenvolupa el joc. A continuació es pretén introduir l'eina amb l'objectiu que qualsevol persona que no hagués sentit a parlar amb anterioritat de la plataforma sigui capaç de seguir el treball i el seu desenvolupament.

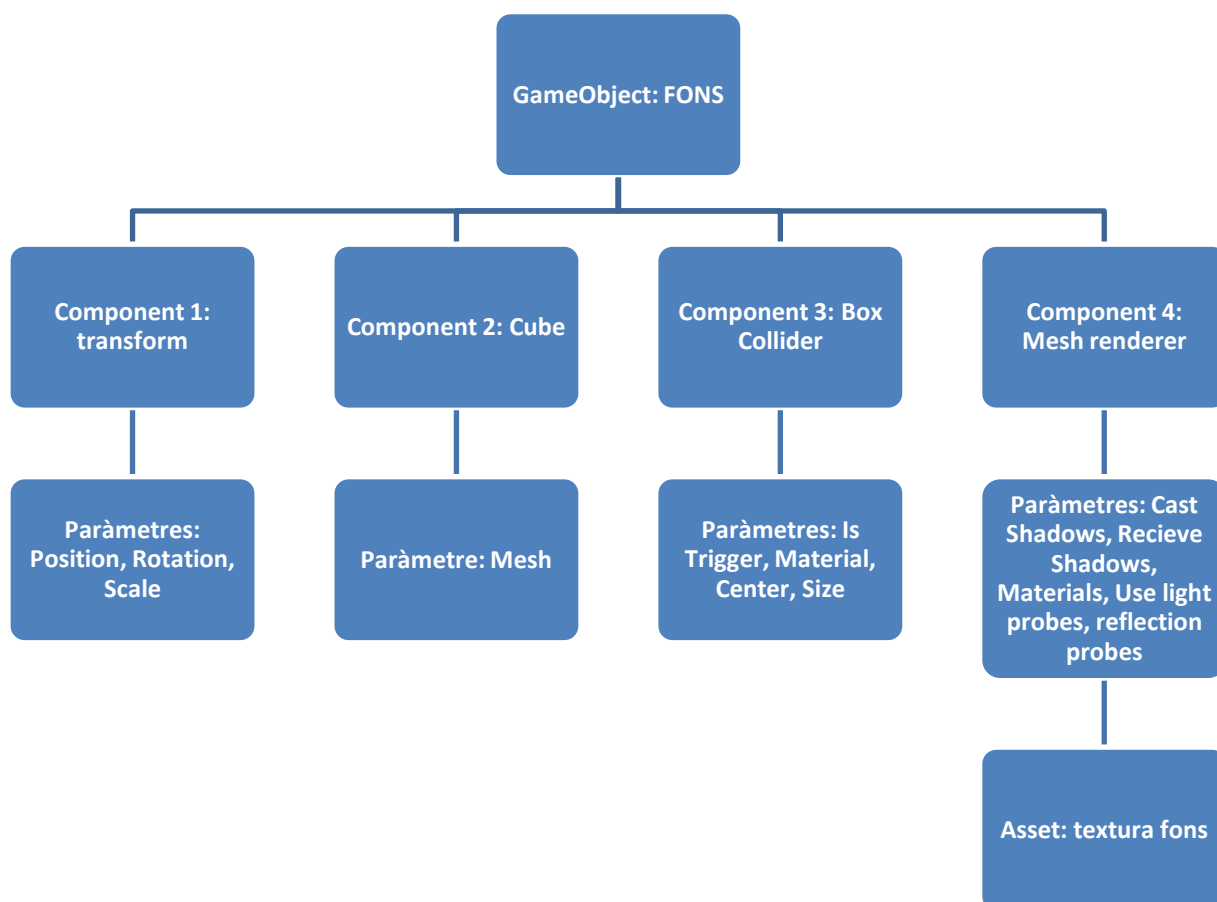
Unity és un motor de videojoc multi-plataforma, creat per Unity Technologies, que està disponible per Microsoft Windows OS i permet desenvolupar jocs per un nombre elevat de plataformes com ara Linux, Play Station, Wii o PC amb Windows, que serà la més utilitzada en aquest cas.

Per a entendre el funcionament de la plataforma cal conèixer en primer lloc l'estructura, els components i objectes dels que depèn qualsevol tipus d'aplicació. Una aplicació pot estar formada per una escena o per més d'una escena, essent aquestes escenaris de joc en els que s'introdueixen tots els objectes i on transcorre l'acció. La plataforma compta amb diverses categories d'objectes, detallades a continuació:

- **GameObjects:** són els elements que s'utilitzaran per desenvolupar l'aplicació, cadascun dels quals emmagatzema al seu interior components que el distingeixen dels altres. Un GameObject pot ser una càmera, un personatge o una llum, i pot ser visible, conformant el conjunt d'elements que es veuen a la pantalla, o bé invisibles, utilitzats únicament per introduir el comportament del sistema o altres components.
- **Components:** propietats que s'atorguen als GameObjects i que contenen diversos paràmetres que modifiquen i caracteritzen el component.
- **Script:** dóna les instruccions i defineix el comportament dels objectes de l'escena. És una llista d'instruccions escrites en un llenguatge informàtic, en aquest cas en C#.
- **Assets:** recursos que poden ser imatges, textures, sons o scripts que s'inclouen dins dels paràmetres dels components.



Per a una millor comprensió s'analitza l'exemple de l'objecte d'un joc que actua de fons de pantalla.



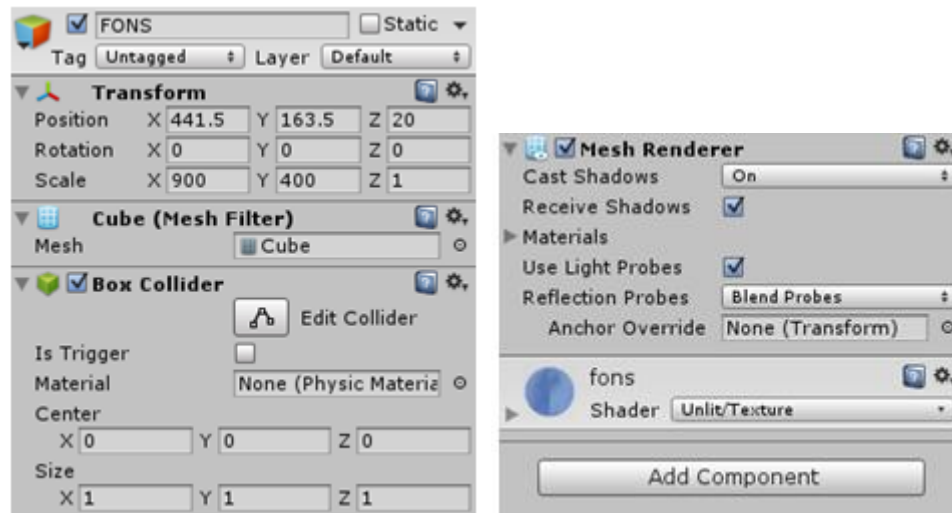


Fig. 7.4. Estructura de l'objecte Fons

El llenguatge d'escriptura C# està destinat a la programació d'objectes i a la programació de recursos gràfics, que té una sintaxi que deriva del llenguatge C i que s'hi assembla força. Aquest és un entorn de programació nou per a l'autora del treball i per aquest motiu s'ha necessitat l'ajuda de diverses fonts d'informació com tutorials d'Internet i la documentació pròpia de la plataforma de Unity.

La interfície gràfica de Unity està formada per diverses zones de treball, que es poden distribuir de la manera que l'usuari desitgi [3]. Tot seguit es fa una breu descripció d'aquestes àrees de treball i de la funcionalitat de cadascuna d'elles.

- **Scene View (Vista d'escena):** permet editar l'escena de manera visual, moure objectes de lloc, incloure'n de nous i seleccionar-los.
- **Game View (Vista de joc):** mostra com es veurà el joc en executar-se i permet jugar-hi a mesura que es va desenvolupant l'aplicació.
- **Hierarchy View (Vista de jerarquia):** mostra la llista dels objectes presents a l'escena de treball, ordenant-los per data de creació i en ordre jeràrquic si n'hi ha que estiguin inclosos dins d'un altre. L'ordre dels objectes de la llista és modificable per l'usuari.
- **Project View (Vista d'inspector):** facilita la visualització de la carpeta d'assets disponibles per fer servir en el joc.
- **Inspector View (Vista d'inspector):** mostra una llista de totes les propietats que se

li han adjudicat a un objecte, permet afegir i treure components i modificar els assets.

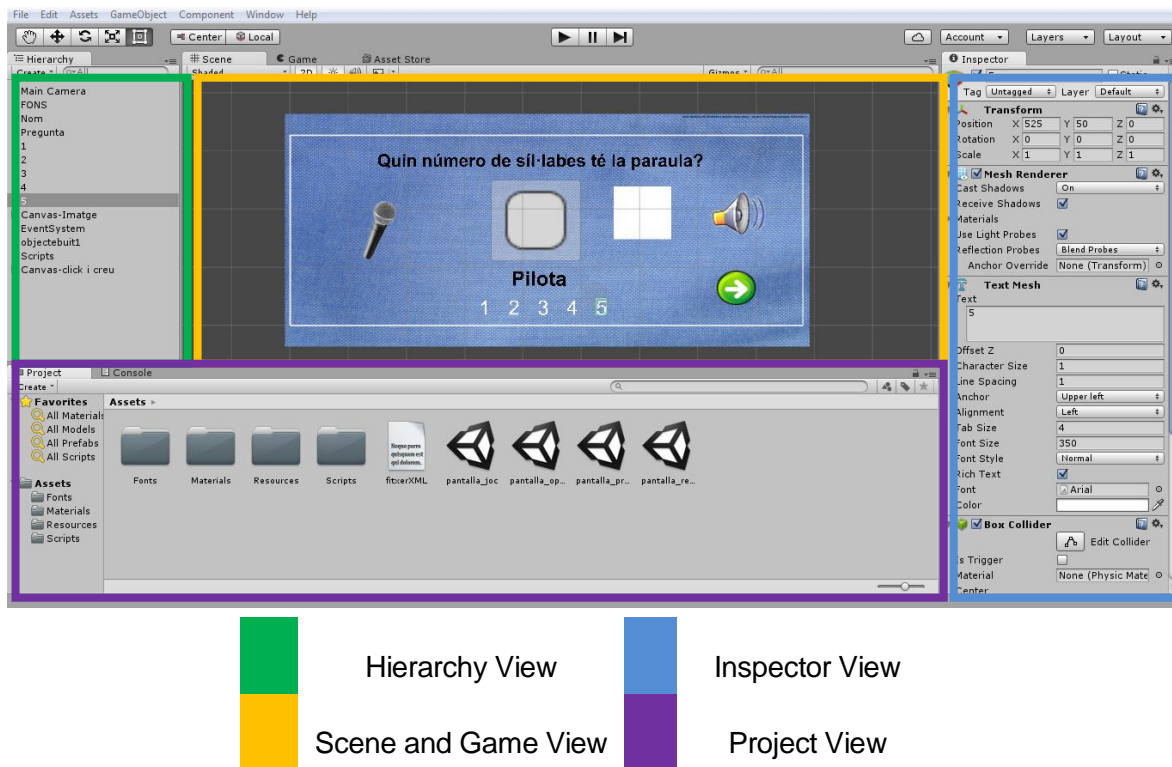


Fig. 7.5. Aspecte del motor de joc

## 7.1.2. Entorn integrat de desenvolupament

L'entorn de desenvolupament integrat (IDE) és una aplicació de software que facilita uns serveis integrats que tenen com a objectiu ajudar al programador en el desenvolupament del programa. Per a tal fi consta de varies funcions: editor de codi, compilador, intèrpret i un programa de disseny d'interfície gràfica.

L'entorn integrat en el que es desenvolupa aquesta aplicació és el MonoDevelop, que és un entorn gratuït i lliure que dissenyat bàsicament per a C#. Aquest entorn permet controlar els objectes de la pantalla a partir dels scripts i definir-ne el seu comportament, donant les ordres necessàries amb llenguatge C#.

### 7.1.2.1. Scripts

Com s'ha vist anteriorment, el comportament dels objectes de l'escena es controla mitjançant components que hi estan associats. Tot i que Unity ja porta inclòs un ampli ventall de components predefinits, et permet crear els teus propis components per poder interaccionar amb els elements de l'escena, modificar les propietats amb el pas del temps i

respondre a les entrades del jugador. Aquests components propis reben el nom de scripts i es creen des de dins de Unity, fent servir per defecte l'entorn integrat MonoDevelop.

Un script fa les seves connexions amb el funcionament intern de Unity implementant una classe que deriva de la classe MonoBehaviour, comentada a continuació. És a dir, cada vegada que es fa un script automàticament es genera una classe que deriva de la classe MonoBehaviour i que contindrà tota les comandes que defineixen el comportament de l'objecte al que anirà associat l'script.

#### 7.1.2.2. Classe MonoBehaviour

Una classe permet crear nous components que es poden adjuntar als GameObjects. Cada vegada que s'adjunta un component a un objecte es crea una nova instància de l'objecte definit per la classe. El nom de la classe s'especifica en el moment en que es defineix el nom de l'script.

La classe MonoBehaviour és la classe base de la programació amb Unity de la que deriven tots els scripts, amb les seves corresponents classes. Aquesta proporciona una llista de les funcions que es poden emprar en els scripts adjunts a GameObjects, com la funció Start() o Update que es comentaran més endavant.

Per a estalviar feina a l'hora de l'escriptura del codi es poden definir namespaces a la part superior de l'script. D'aquesta manera es crida la classe predefinida per Unity i així cada vegada que es faci referència a aquesta no caldrà mencionar-la ni afegir-la com a prefix.

Per tot el comentat anteriorment, la definició d'una classe (*exemple*) i la crida de les classes predefinides que s'empraran queda de la següent manera:

```
using UnityEngine;

using System.Collections;

using System;

public class exemple : MonoBehaviour {

}
```

Les classes UnityEngine i System.Collections estan cridades per defecte en tots els scripts de l'aplicació.

La classe MonoBehaviour té algunes funcions definides per defecte, que són molt útils per al desenvolupament de qualsevol aplicació i que han estat molt utilitzats en aquest treball. Els mètodes principals que té definits per defecte la classe són els següents:

- **Awake():** funció que es crida quan s'està carregant la instància de l'script.
- **Start():** la funció es crida quan l'script s'està carregant i s'executa abans que cap altre funció.
- **OnMouseDown():** la funció es crida i s'executa quan el jugador prem el botó del ratolí sobre un objecte o un component Box Collider.

## 7.2. Processament de la informació

El processament de la informació en l'aplicació consisteix bàsicament en importar tota la informació necessària per a l'execució de les tasques d'un arxiu extern. S'ha comentat anteriorment la necessitat de disposar d'un mètode de processament de la informació que permeti la modificació de les paraules amb les que es treballarà i la seva ampliació per a què el logopeda pugui adaptar l'aplicació a l'usuari incorporant el vocabulari que trobi més oportú.

Cal també tenir en consideració que el seguiment de l'evolució del pacient era una prioritat i un requeriment que havia de complir l'aplicació. Tot i així, no s'ha incorporat un sistema d'emmagatzematge extern ja que es creu que el llistat de resultats acumulats podria ser feixuc i difícil d'entendre amb el temps. L'aplicació el què fa és permetre la visualització dels resultats un cop finalitzat el joc i fer accessible la informació fins que s'inicia un nou joc.

A més, l'aplicació permet enregistrar les freqüències obtingudes dels àudios, tant els models com l'obtingut a partir del micròfon. Aquest enregistrament es duu a terme en fitxers TXT ja que són fàcils d'escriure i llegir.

Per a l'emmagatzematge de tota la informació que permetrà el desenvolupament del joc, cal dissenyar una base de dades ben organitzada que contingui dades de la paraula, imatge, so i nivell de cadascuna de les pantalles del joc. S'ha triat el format XML com a format de la base de dades ja que s'ha cregut que era un dels més senzills de tractar per a gent que no es dediqui a la informàtica, com són els logopedes.

A continuació es detallarà les característiques, el mètode de funcionament i l'estructura dels dos models de fitxers que s'empren en l'aplicació.

### 7.2.1. Fitxer XML

El fitxer XML (eXtensible Markup Language) s'utilitza per emmagatzemar dades de forma llegible, dóna suport a les bases de dades i és útil quan diverses aplicacions s'han de comunicar entre sí.



La raó per la qual s'ha decidit utilitzar aquest tipus de fitxers rau en els següents motius:

- És extensible: un cop s'ha dissenyat l'arxiu, aquest es pot modificar i ampliar per mitjà de l'addició de noves línies de text en les que s'inclouen noves etiquetes.
- L'estructura del fitxer és senzilla i fàcil de processar, fet que facilitarà la tasca del logopeda a l'hora d'ampliar i modificar el seu contingut.
- Permet comunicar aplicacions de plataformes diferents sense que tingui importància l'origen de les dades.
- Converteix les dades en informació ja que se li afegeix un significat concret en un context concret.

Aquesta tecnologia presenta la informació de manera estructurada, és a dir, composta per parts ben definides i que aquestes parts es composin alhora d'altres parts. D'aquesta manera s'obté una estructura d'arbre, els elements del qual s'anomenen etiquetes, i contenen informació clara i definida.

Un document XML consta de dues parts: el pròleg i el cos del document, així com d'un text d'etiquetes que contenen tota la informació.

- Pròleg: Conté una declaració XML, que es la part que declara el document com a XML i està situada a la capçalera, una declaració de tipus de document i un o més comentaris i instruccions de processament, com per exemple: `<?xml version="1.0" encoding="utf-8"?>`, que s'utilitza per a què el fitxer sigui capaç d'interpretar els accents i altre simbologia no bàsica.
- Cos: Conté les parts en què es divideix la informació, que s'anomenen elements i es defineixen amb etiquetes. Les etiquetes es defineixen amb el nom que es desitgi delimitat pels signes `<>`, per exemple `<nom>`. Quan les etiquetes contenen tota la informació que es vol cal tancar-les, de manera que l'estructura seria similar a la següent en què primer s'obre l'etiqueta, després s'afegeix tota la informació desitjada i finalment es tanca delimitant-la amb els signes `</>`, per exemple `</nom>`.

`<exemple>`

`<individu1 edat="59" ciutat="Barcelona" professió="enginyer">`

`</exemple>`

El cos l'encapçala obligatòriament un element arrel, però a partir d'aquest se'n poden derivar altres. Totes les etiquetes poden tenir atributs, que incorporen característiques a l'element,

diferenciant-lo així dels altres. El valor dels atributs ha d'anar indicat entre cometes ("" ). L'estructura que es segueix per a la definició d'elements amb atributs segueix és similar a la de l'exemple següents:

```
<individu1 edat="59" ciutat="Barcelona" professió="enginyer">  
</individu1>
```

En aquest exemple l'element és individu1, que incorpora com a atributs l'edat, la ciutat i la professió.

Finalment el fitxer que s'ha utilitzat és el següent:

```
<?xml version="1.0" encoding="utf-8"?>  
<JOC>  
  <dadesCAT>  
    <imatge0 nom="pilota" foto="pilota" so="pilotaSo" silabes="3" nivell="2">  
    </imatge0>  
  </dadesCAT>  
  <dadesCAST>  
    <imatge0 nom="pelota" foto="pilota" so="pelotaSo" silabes="3" nivell="2">  
    </imatge0>  
  </dadesCAST>  
</JOC>
```

L'arxiu es subdivideix en les següents parts:

- Capçalera, que com s'ha comentat anteriorment incorpora la línia que facilita la interpretació de símbols que no s'utilitzen habitualment en codis de programació, com per exemple els accents.
- L'arxiu conté un element arrel <JOC>, que incorpora tota la resta d'elements.
- L'element arrel es subdivideix en dos elements fills <DadesCAT> i <DadesCAST>, que serveixen per emmagatzemar cadascun d'ells la informació del joc en català i en castellà respectivament.
- Dins dels elements que separen es informacions per idiomes es poden trobar les

imatges que es visualitzaran en el transcurs del joc. Cadascuna de es imatges té cinc atributs que la caracteritzen: nom, foto, so, silabes i nivell. Definint el valor que prenen aquests atributs es defineix la paraula per a poder executar correctament el joc.

### **7.2.2. Fitxers TXT**

Els fitxers de text guarden les dades com a caràcters i per això requereixen d'un sistema de visualització com un editor de text (com per exemple el notepad). El mètode d'escriptura en aquests fitxers és variat, podent-se utilitzar el llenguatge binari o altres per codificar el text, fet que els fa molt útils per a l'emmagatzematge de dades.

El mètode d'escriptura és molt senzill i la representació del contingut en forma de lleres, números i signes de puntuació en facilita la lectura. Bàsicament per aquest motiu s'ha triat fer servir aquest sistema per a l'emmagatzematge de les dades de les freqüències ja que la pròpia plataforma de Unity disposa de mètodes d'escriptura i lectura d'aquest tipus de documents i, a més, són molt senzills de tractar.

## 8. Base teòrica del reconeixement de veu

Com s'ha esmentat als objectius (ap. 3.1), una de les finalitats del projecte és l'implementació de reconeixement de veu en l'aplicació per a que els pacients puguin desenvolupar les seves habilitats de la parla. Per aquest motiu, s'ha fet un estudi dels possibles mètodes de reconeixement de veu que es podrien emprar i que estan en ús en l'actualitat.

A continuació es fa un recull dels mètodes vigents de reconeixement de veu, que són molt diversos. S'ha pres com a representatius: el sistema de reconeixement de veu de Google, el mètode que empra el Vivo, el Dynamic time warping (DTW) i el Unity System Speech Recognition.

A partir d'aquesta cerca es fa un estudi de les diverses opcions per tal de verificar quina podria adequar-se més al treball.

Per tal de poder entendre el funcionament de les opcions de reconeixement de veu convé tenir clars alguns conceptes que hi estan directament relacionats i que es detallen a continuació [5], [7].

Un senyal acústic és un senyal analògic, és a dir, és un flux continu d'ones sonores i silencis. En aquest treball es farà servir el senyal per a la identificació i descripció dels atributs de la parla, ja que aquests són necessaris per al reconeixement de veu.

Els sons produeixen moviments en les molècules de l'aire, conformant patrons regulars i prolongats que es poden representar com a ones sinusoïdals (per a sons purs, sense soroll). Hi ha quatre característiques que permeten definir l'anàlisi acústic:

- **Freqüències:** nombre de vibracions o cicles del to per segon. Exemple: 100 cicles/segon=100Hz
- **Amplitud:** fisiològicament representa el moviment del timpà i analíticament s'interpreta com el volum del so, que reflexa la quantitat d'aire que es mou i es mesura en decibels dB
- **Estructura Harmònica (to vs soroll):** la parla no és un so pur sinó que és una combinació de múltiple freqüències que es representen de forma complexa. Les ones sonores de la veu estan formades per patrons simples que són rics en freqüències secundàries, amb estructures cícliques i acícliques. Les ones acícliques formen part de tots els fonemes i s'anomenen soroll.

Per esbrinar la identitat dels fonemes cal estudiar la freqüència i característiques dels patrons o bé estudiar el canvi en els **formants [6]**. Aquests es poden definir com les bandes de freqüències secundàries que es troben combinades amb una freqüència dominant o fonamental, que contribueixen al timbre del so i que juguen un paper important en la distinció d'un fonema respecte un altre.

- **Ressonància:** Es defineix com la capacitat que té una font vibrant de so de causar la vibració d'un altre objecte. Per exemple, la boca i el nas són cambres de ressonància i amplifiquen les bandes de freqüències contingudes en el so que es genera a les cordes vocals.

Pel que fa a la metodologia de reconeixement tots els sistemes estan destinats a complir amb tres fases:

- **Preprocessament:** Converteix l'entrada de la veu en una forma que el reconeixedor pugui processar.
- **Reconeixement:** Identifica el què s'ha dit.
- **Comunicació:** Envia el què s'ha reconegut al sistema que ho requereixi.

Tots ells a més a més estan conformats pels següents components:

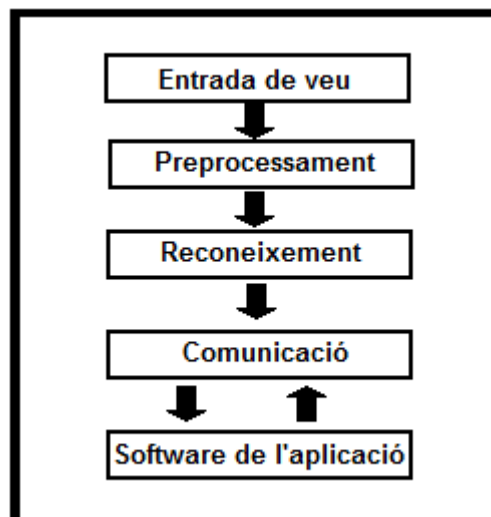


Fig. 8.6. Esquema dels components d'un sistema de reconeixement de veu

### 8.1.1. Sistema de reconeixement de veu de Google

Els sistemes de reconeixement de veu actuals tenen un processament complex, que consta de diverses fases les quals es detallen breument a continuació:

1. Captura de sons a partir d'un micròfon.
2. Digitalització del so a través d'un convertidor analògic-digital (A/D)
3. Les dades digitals es representen en un espectrograma fent servir la tècnica matemàtica de la transformada de Fourier (FFT, Fast Fourier Transform). Les dades es processen i s'analitzen per trobar els components que les caracteritzen
4. Comparació de les dades característiques amb un diccionari fonètic per identificar què és el que probablement s'ha dit. Mai es pot assegurar al 100%.
5. Utilització d'un model de llenguatge, que defineix quines combinacions de paraules són més probables i quines no. Per exemple, després del mot "exemple" és molt més probable que hi vagi un altre com "de", "per", "bo" i "dolent" que un com "taula" o "tren".
6. Determinació de quina és la paraula més probable que correspongui al so captat pel micròfon. Usualment es fa servir un algoritme anomenat Viterbi algorithm.
7. Correcció dels errors més habituals que comet el sistema de reconeixement gràcies a l'aprenentatge, mitjançant un feedback.

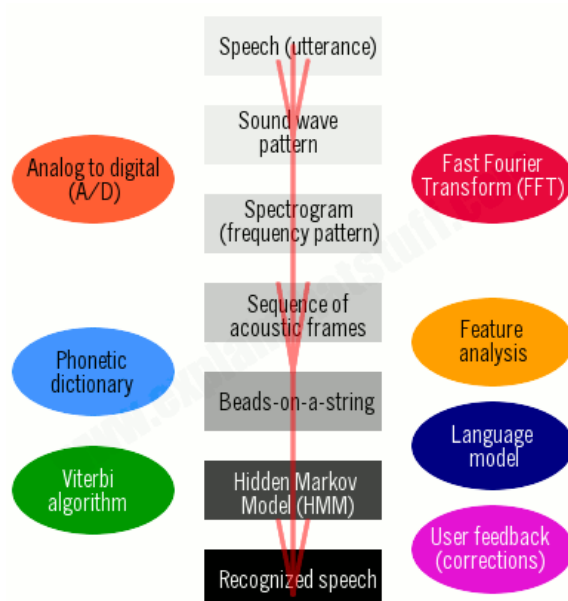


Fig. 8.7. Esquema del funcionament dels sistemes de reconeixement actuals

El sistema de reconeixement de veu de Google fa servir aquesta metodologia, però com s'ha pogut comprovar aquest mètode escapa de l'abast d'aquest projecte ja que no es disposa de prou temps com per dur a terme tots aquests passos de manera correcta i a més tampoc es disposa de prous coneixements en la matèria.

S'ha considerat també que aquest mètode no era necessari ja que en aquest cas només es volen reconèixer paraules aïllades i no cal implementar una tecnologia tant complexa com la descrita anteriorment per a dur-ho a terme.

Per tant, aquest mètode ha quedat descartat per a aplicar-lo en aquest treball, encara que se sap que és el més eficient i efectiu ja que és amb el que treballen les grans potències del sector en l'actualitat.

### 8.1.2. Vivo

El Vivo és una aplicació que està desenvolupada pel mateix projecte que l'aplicació Vocaliza (ap. 4.2), el qual s'anomena projecte Comunica. Aquest s'encarrega de l'àrea dels jocs per a la rehabilitació logopèdica i està integrat per membres de la *Universidad de Zaragoza*.

Aquesta aplicació pretén mostrar en temps real la producció de sons del pacient i permet diferenciar entre les vocals a partir dels valors de les freqüències dels seus dos primers formants. Així doncs, dels sons només en considera i analitza les vocals.

El reconeixement de vocals és força més senzill partint de les freqüències dels dos primers formants ja que cada rang de freqüències del gràfic que es mostra a continuació delimita cadascuna de les vocals. Es pot observar que només hi ha cinc vocals ja que es corresponen als sons vocàlics que hi ha en castellà.

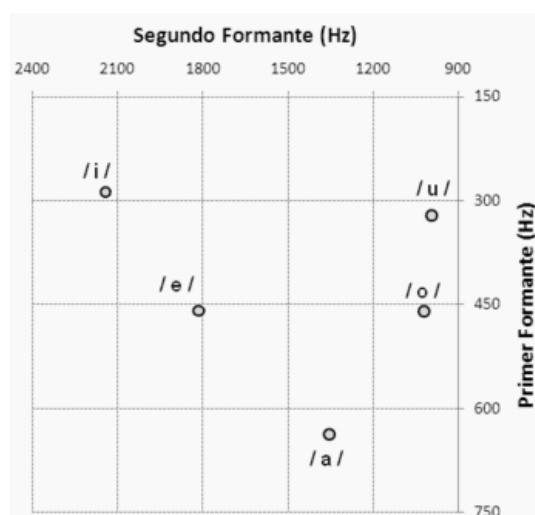


Fig. 8.8. Representació del sons vocàlics en funció dels formants

L'aspecte de l'aplicació és el que s'observa a la imatge següent. Està dividida en diverses zones. A la part superior hi ha opcions per a que l'usuari defineixi la seva constitució ja que en funció d'aquesta l'espectre de veu varia. A mà dreta es troba la visualització de l'espectre de veu i la indicació dels valors d'intensitat, pitch, freqüències dels dos primers formants i decibels. I, finalment, hi ha un visualitzador de les vocals que consisteix en una representació de les freqüències secundàries en un gràfic on l'eix x representa les freqüències del primer formant (F1) i l'eix y les freqüències del segon formant (F2). En aquesta àrea també hi ha un requadre que permet la visualització de la vocal que el programa està interpretant quan els punts del gràfic es troben dins d'alguna de les regions que estan delimitades per cercles.

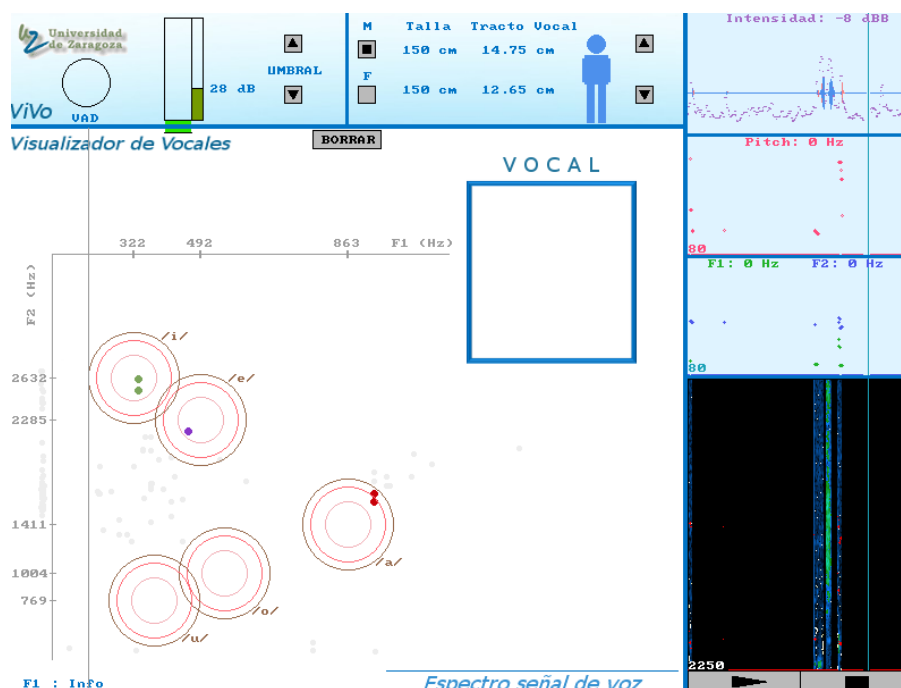


Fig. 8.9. Visualització de la pantalla del Vivo

A partir de la informació anterior es pot concloure que aquest seria un bon mètode per a la implementació del reconeixement de veu en aquesta aplicació ja que en aquesta tasca només s'ha de distingir entre els cinc primers números decimals enters (u, dos, tres, quatre, cinc). Es pot observar que el nom d'aquests nombres contenen síl·labes tòniques amb vocals diferents, fet que indica que aquest pot ser un mètode efectiu per a la seva distinció.

Tot i així, l'obtenció de les freqüències secundàries que conformen els formants no és una tasca senzilla i després d'un primer intent es va decidir emprar una altra metodologia de reconeixement de veu.



### 8.1.3. Dynamic time warping (DTW)

El sistema DTW no és un sistema de reconeixement per sí sol sinó que consisteix en un algoritme que es pot utilitzar per a reconèixer la veu. L'algoritme s'utilitza per mesura la semblança entre dues seqüències temporals, que poden variar en temps i velocitat. Per exemple, si es volguessin trobar la semblança entre les trajectòries de dues persones caminant es podria executar l'algoritme i funcionaria encara que una persona anés a més velocitat que l'altre o hi haguessin acceleracions i desceleracions.

Aquest algoritme pot analitzar qualsevol dada que es pugui convertir en una seqüència però majoritàriament s'ha utilitzat per a la comparació de seqüències d'àudio, vídeo i dades de gràfics. Una aplicació molt coneguda és la del reconeixement de veu tot i que el sistema ja ha quedat obsolet i les grans potències fan servir mètodes més desenvolupats.

El codi original de l'algoritme és el següent, on  $s$  i  $t$  són dues llistes de mida  $n$  i  $m$  respectivament i on  $d(x, y) = |x - y|$  és la distància en valor absolut entre dos valors.

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
    DTW := array [0..n, 0..m]

    for i := 1 to n
        DTW[i, 0] := infinity
    for i := 1 to m
        DTW[0, i] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j], // insertion
                                         DTW[i, j-1], // deletion
                                         DTW[i-1, j-1]) // match

    return DTW[n, m]
}
```

Fig. 8.10. Codi extret de la referència [4]

La funcionalitat d'aquest algoritme és trobar la distància total entre dues llistes fent una suma de la diferència entre els valors dels nombres de les llistes que es trobin en la mateixa posició i la diferència amb els dels seus nodes veïns.

Finalment, a partir de la informació trobada s'ha considerat que aquest era un bon mètode per a aplicar en aquest treball. Les dues llistes que es compararan seran les llistes de freqüències de l'àudio d'entrada del micròfon (els sons que emetrà el pacient) i dels àudios que faran de model (àudios enregistrats prèviament amb el contingut de les paraules corresponents en números 1,2,3,4,5 en català i castellà).

#### **8.1.4. Unity System Speech Recognition [13]**

La plataforma Unity conté una classe que actua com a sistema de reconeixement de veu propi i que s'anomena System Speech Recognition. Aquesta classe conté funcions que permeten el reconeixement de veu i la seva correcta interpretació.

Aquest sistema seria perfecte si existís en els idiomes requerits en aquesta aplicació que són el català i el castellà, però de moment només està implementat en anglès, que és l'idioma en què està feta tota la plataforma Unity.

El sistema, a més, presenta algunes limitacions ja que s'ha de pronunciar les paraules correctament per a que el sistema sigui capaç d'interpretar-les, fet que no és convenient en aquesta aplicació ja que va adreçada a infants amb patologies de la parla.

## 9. Estructura del joc

En aquest apartat es dona tota la informació necessària per entendre en quines parts està estructurat el joc, com s'han desenvolupat i quin funcionament tenen aquestes a dins de l'aplicació.

Per organitzar l'explicació de l'estructura s'ha subdividit el conjunt de l'aplicació en escenes, que, com s'ha comentat a l'ap. 7.1.1, són els diferents escenaris en els quals transcorre l'acció. Així doncs, hi ha 4 subgrups, que corresponen a les quatre escenes que té el joc: pantalla\_principal, pantalla\_opcions, pantalla\_joc i pantalla\_resultats.

Per a cada escena s'expliquen els següents aspectes, seguint l'ordre de dependència que s'ha comentat a l'ap. 7.1.1:

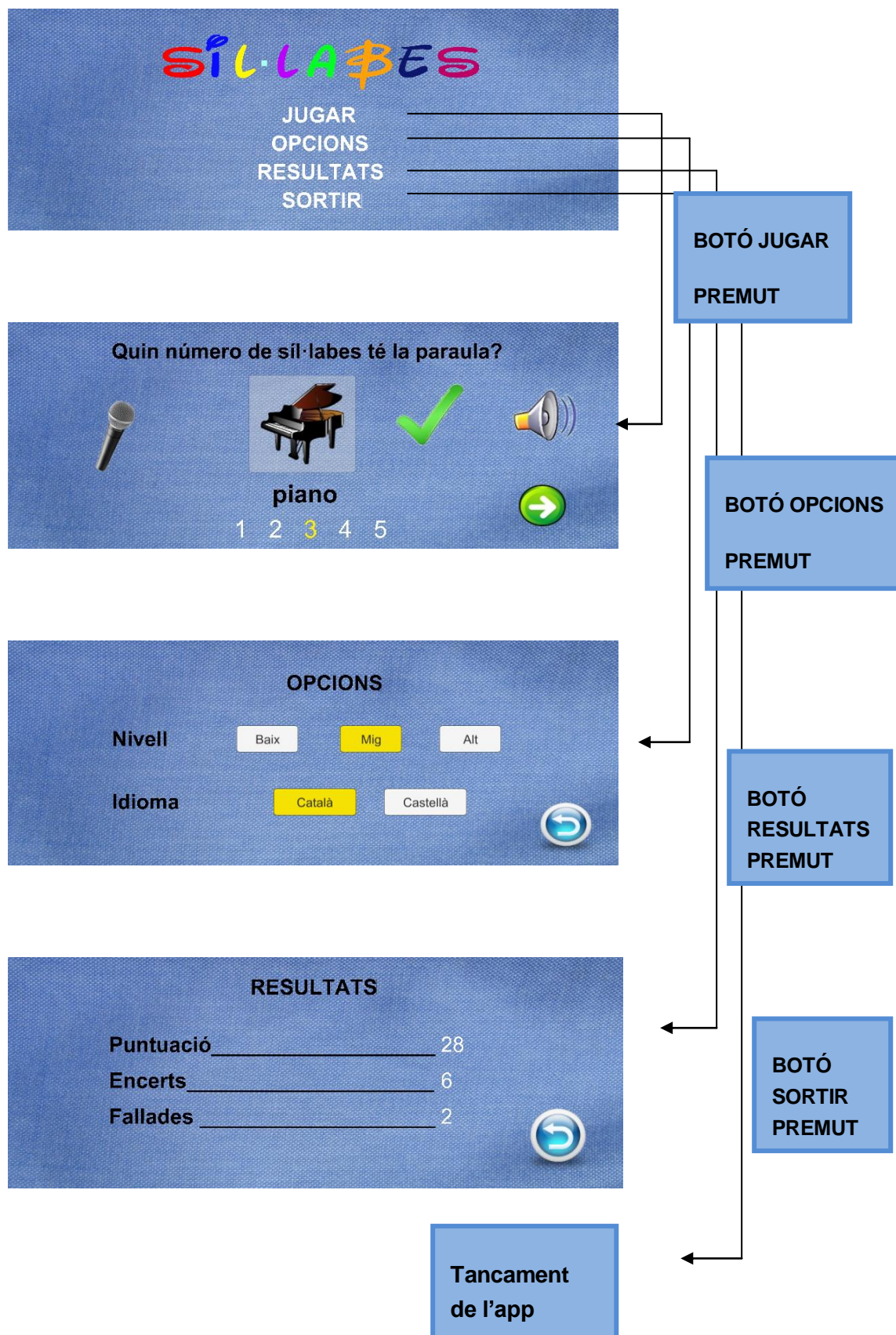
- Funcionament
- GameObjects
  - Components i assets
  - Scripts

### 9.1. Pantalla\_principal

La pantalla\_principal és l'escena d'inici del joc, que fa de base per a que l'usuari pugui definir les condicions en les que vol iniciar el joc, l'instant en que el vol començar, pugui visualitzar els resultats i aturar el funcionament de l'aplicació.

L'aplicació està pensada per a que el logopeda en porti el control i defineixi les opcions en funció de les necessitats i habilitats del pacient, per tant serà principalment el logopeda el que tindrà contacte amb aquesta pantalla.

### 9.1.1. Funcionament



Com es pot observar en l'esquema anterior la pantalla principal permet accedir a les diverses escenes del joc en funció de quin botó es premi.

### 9.1.2. GameObjects

#### 9.1.2.1. Main Camera (càmera principal)

És un objecte principal en totes les escenes de l'aplicació que permet controlar les característiques bàsiques de visualització de la pantalla. És el dispositiu a través del qual l'usuari veu el joc. Els punts de la pantalla es defineixen mitjançant píxels, essent l'extrem inferior esquerra l'origen (0,0).

- Components i assets

La càmera conté 5 components per defecte: Transform, Camera, GUI Layer, Flare Layer i Audio Listener. Aquests elements permeten renderitzar els objectes de la interfície gràfica, canviar la posició de l'objecte, orientar la projecció de la imatge, modificar els efectes de llums i reproduir sons de l'escena (només pot haver-hi un Audio Listener a l'escena).

- Scripts

La càmera té associat un script que permet adaptar la dimensió de l'escena i la posició dels objectes que conté a la dimensió de la pantalla del dispositiu que s'utilitzi. Aquest script s'anomena AspectUtility i permet obtenir la màxima relació entre l'alçada i l'amplada sigui quin sigui el tipus de resolució. La relació d'aspecte (entre alçada i amplada) que s'ha triat és 16:9 perquè és la relació d'aspecte més usual en pantalles d'ordinador. En cas de voler adaptar l'aplicació a mòbils s'hauria de reajustar la relació d'aspecte estudiant les pantalles d'aquests dispositius i veient quina és la relació d'aspecte més comuna.

#### 9.1.2.2. Fons

El fons és una superfície de color sobre la que es representen les figures i objectes. En aquest cas, s'ha emprat un objecte del tipus 3D Object, Cube, dotant-lo de majors dimensions en l'eix x i y i deixant-la únicament amb una profunditat de 1.

Cal destacar que aquest objecte serà comú en totes les pantalles del joc perquè d'aquesta manera s'aconsegueix donar uniformitat al joc.

- Components i assets

Un cub té els components següents per defecte: Transform, Cube (mesh filter), Box Collider

i Mesh Renderer, que permeten canviar de posició l'objecte, definir un mallat i generar una àrea per a habilitar la interacció de l'objecte amb el ratolí. Al Mesh Renderer se li ha afegit un asset que consisteix en una textura estreta de la xarxa, que és força discreta per no desviar el focus d'atenció del pacient.

S'ha decidit no incorporar cap element del tipus AudioSource ja que les músiques poden distreure l'usuari o poden molestar a possibles usuaris tercers que es trobin a la vora.

- Scripts

El fons no porta cap script adherit ja que aquest té una funció visual, no ha de tenir cap comportament en concret ni ha d'interactuar amb l'entorn.

#### 9.1.2.3. Títol

El títol es troba situat a la part superior de la pantalla i serveix per designar l'aplicació. El nom que té aquest aplicació és *Síl·labes* ja que és clarificador del contingut de l'aplicació. Està format per 9 GameObjects, un per a cadascuna dels caràcters que el conformen. Aquests GameObjects són del tipus 3D Object, 3DText.

- Components i assets

Els GameObjects del tipus 3D Object, 3D Text porten per defecte els components: Transform, Mesh Renderer i Text Mesh. Els components Transform s'han usat per definir en quina posició es troba cadascun dels objectes dins de l'escena, el Mesh Renderer s'ha deixat en els valors que portava per defecte i els Text Mesh s'han utilitzat per definir els caràcters que es volia representar en cada GameObject d'aquest tipus, definir-ne la mida de lletra i el color.

Als components Text Mesh se'ls ha afegit un asset corresponent a un tipus de font que s'ha extret de la xarxa. S'ha considerat que la font emprada era adient per a nens i nenes ja que és força atractiva per a aquest públic i, a més, pintant-ne una de cada color ha quedat un títol vistós.

- Scripts

Els GameObjects que conformen el títol no porten cap script adherit ja que aquest té una funció merament informativa i visual, no ha de tenir cap comportament en concret ni ha d'interactuar amb l'entorn.

#### 9.1.2.4. Jugar, opcions, resultats i sortir

Els objectes de l'escena que mostren les paraules "jugar", "opcions", "resultats" i "sortir" són



objectes en 3D del tipus text en 3D. Aquests objectes realitzen la funció de botons, és a dir, quan els prems o bé carreguen altres escenes del joc o bé tanquen l'aplicació. Les pantalles que es carreguen són: pantalla\_joc, pantalla\_opcions i pantalla\_resultats en prémer els botons, jugar, opcions i resultats respectivament. El tancament de l'aplicació es realitza doncs prement sobre el text de sortir.

- Components i assets

Com s'ha comentat anteriorment, aquest tipus d'objectes tenen per defecte els components: Transform, Mesh Renderer i Text Mesh. El component Transform s'ha usat per a la col·locació dels objectes en l'espai de l'escena, el component Mesh Renderer s'ha deixat amb els valors per defecte i el Text Mesh ha servit per modificar la grandària de la lletra i el tipus.

A més, s'ha afegit un component Box Collider, delimitant el contorn del text per tal de poder interaccionar amb el ratolí.

- Scripts

Els quatre objectes de text tenen associat un script anomenat Canvi\_color, que s'encarrega de posar a groc el text en passar-hi per sobre i tornar-lo al color original, blanc, quan se surti de la seva àrea a no ser que s'hagi premut el botó esquerra del ratolí per a seleccionar el text.

L'script Canvi\_color té dues funcions:

- OnMouseEnter(): és una funció que executa el seu contingut quan el ratolí entra a l'àrea delimitada per l'objecte al que va associat l'script. En aquest cas, quan s'accedeix a l'àrea el que fa és convertir el paràmetre Color del Text Mesh en color groc enlloc de blanc, que és el seu color natural.
- OnMouseExit(): és una funció que executa el seu contingut quan el ratolí surt de l'àrea delimitada per l'objecte al que va associat l'script. En aquest cas, quan se surt de l'àrea el paràmetre Color del Text Mesh es converteix a blanc.

A més a més, cadascun dels objectes porta associat un altre script anomenat Boto\_jugar, Boto\_opcions, Boto\_resultats o Boto\_sortir que permet que es realitzi una transició cap a l'escena que correspongui. Aquests scripts són molt simples i consisteixen en una única funció:

- OnMouseDown(): és una funció que executa el seu contingut quan es prem amb el botó esquerra del ratolí l'espai que delimita l'objecte al que va associat l'script. El

contingut de la funció és la transició de l'escena principal a l'escena que es desitgi. El mètode que s'utilitza per a fer la transició és LoadLevel, que forma part de la classe Application i que es crida de la següent manera:

```
Application.LoadLevel("nom de l'escena a la que es vol anar");
```

En el cas del botó sortir dins de la funció OnMouseDown() s'hi trobarà el mètode Quit() de la llibreria Application, que es crida com es mostra a continuació:

```
Application.Quit();
```

## 9.2. Pantalla\_opcions

L'escena d'opcions permet triar els paràmetres comentats a l'apartat 6.2.1 i, per tant, és una de les més importants del joc ja que dóna la capacitat al logopeda d'adaptar el joc a les capacitats que tingui el pacient. S'han introduït dos paràmetres diferents, que són el nivell de dificultat i l'idioma, com es pot veure a la imatge següent.



Fig. 9.11. Visualització de la pantalla opcions

- **Nivell:** Està clar que cada persona té un ritme d'aprenentatge i que no tots els pacients tenen el mateix nivell, per això s'ha considerat oportú afegir un paràmetre que pugui adaptar el nivell de dificultat (baix, mig i alt).

A continuació es detalla el criteri que s'ha seguit per determinar el grau de dificultat de les paraules i a quin grup de mots correspon cada nivell.

Criteri: Com més síl·labes té una paraula, més difícil es considera que serà la seva segmentació. Aquest criteri és adaptable al criteri del logopeda i modificable a partir



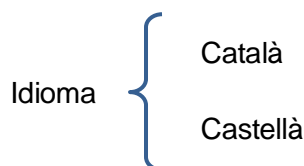
de la base de dades.

Categories:

Nombre de síl·labes	1	2	3	4	5
Baix					
Mig					
Alt					

- Idioma: l'aplicació tenia com a requeriment que es pogués desenvolupar tant en llengua catalana com en llengua castellana ja que a l'hospital la llengua materna dels pacients és diversa. D'aquesta manera poden desenvolupar més l'habilitat de la parla en l'idioma que més utilitzin i inclús practicar amb l'altre per tal de dominar els dos idiomes oficials a Catalunya.

Categories:



La configuració que hi ha per defecte definides pel programa són l'idioma català i el nivell mig. S'ha considerat que aquestes podrien ser les més utilitzades, tot i que no comporta gaire feina el fet d'entrar en aquesta escena i seleccionar altres opcions en cas de que el logopeda així ho vulgui.

Paràmetre	Valor
Nivell	Mig
Idioma	Català

### 9.2.1. Funcionament

Aquesta pantalla està dissenyada per a que el logopeda adapti l'entorn del joc a les capacitats dels usuaris. El funcionament és senzill i intuïtiu ja que les categories es troben ubicades en fila a l'alçada dels paràmetres genèrics que les contenen (visualitzar la imatge 9.11). A més, les categories actuen com a botons, de manera que quan es prem amb el botó esquerra del ratolí sobre d'elles aquestes queden seleccionades. Es pot observar visualment

quina és la selecció que s'ha fet ja que el botó canvia el color del seu fons, posant-se a groc.

## **9.2.2. GameObjects**

### **9.2.2.1. Main Camera (càmera principal)**

Aquest objecte està comentat a l'apartat 9.1.2.1 i els seus components són els mateixos ja que no s'ha incorporat cap component addicional en la càmera d'aquesta escena. També porta incorporat l'script que permet l'adaptació del contingut de l'aplicació a la mida de pantalla del dispositiu que es fa servir, anomenada AspectUtility.

### **9.2.2.2. Fons**

L'objecte que fa de fons de pantalla és exactament igual per totes les escenes. En aquest cas tampoc se li ha afegit cap component addicional i per tant, la seva estructura és idèntica a la que s'ha comentat a l'apartat 9.1.2.2.

### **9.2.2.3. Opcions, nivell i idioma**

L'objecte que mostra el text "OPCIONES" fa de títol de l'escena i introdueix el contingut d'aquesta, mentre que els objectes que mostren els textos "Nivell" i "Idioma" introdueixen els paràmetres variables de joc.

- Components i assets

Tots tres textos estan formats per un objecte del tipus 3D Text, que com s'ha vist anteriorment conté els components: Transform, Mesh Renderer i Text Mesh. Aquest últim és el més important ja que és en el que s'introdueix el fragment de text que es vol visualitzar i en el que es trien aspectes bàsics del text, com ara la mida, el tipus de lletra i el color.

- Scripts

Aquests objectes no tenen associat cap script ni component addicional com un Box Collider perquè no han de comportar-se de cap manera, són figures per ser llegides i donar informació a l'usuari.

### **9.2.2.4. Botons Baix, Mig, Alt, Català i Castellà**

A la pantalla mostrada a la figura 9.11, s'observa la presència de cinc botons del tipus UI Button. Aquests tenen la funció d'executar el seu contingut un cop són premuts. L'ús d'aquests botons és molt habitual en aplicacions i jocs d'ordinador i en aquest treball s'empren en varies ocasions.

Els elements UI Button es col·loquen per defecte en un canvas, que es genera automàticament en agregar un botó a l'escena. Un canvas és un Game Object amb un component canvas que té la funció d'espai abstracte on es col·loquen tots els elements de la interfície d'usuari (UI). En aquest cas se n'han generat dos, un que conté els botons per a les categories de nivell i l'altre per a les categories d'idioma.

- Components i assets

Atès que cada botó ha d'anar dins d'un element anomenat canvas, es procedirà a l'explicació, en primer lloc, de l'estructura dels canvas presents a l'escena i a continuació es detallarà els components que porten incorporats els botons.

Els canvas tenen per defecte els components: Rect Transform, Canvas, Canvas Scaler i Graphic Scaler, que permet posicionar i regular la mida del rectangle que conforma l'àrea de canvas.

Els botons porten per defecte els components: Rect Transform, Canvas Renderer, Image i Button. El primer permet col·locar el botó dins l'espai del canvas. El canvas Renderer permet el renderitzat dels objectes de l'interfície gràfica de l'usuari, el component Image permet agregar un asset en forma d'imatge que es col·locarà a la superfície del botó i el component Button permet definir els colors que prendre el botó en la seva interacció amb el ratolí i mitjançant un script permet també definir la funció que s'haurà d'executar quan es premi aquest.

En aquest cas, s'ha fet que el botó canviï de color quan el ratolí entra a la seva superfície a partir del paràmetre Highlighted Color. També s'ha usat la funció On Click () continguda en el component botó per a definir el comportament d'aquest com es veurà a continuació.

- Scripts

El component Botó porta incorporat un paràmetre que permet seleccionar quina funció s'executarà en el moment en que es premi el botó.

- On Click(): aquesta funció s'executa en el moment que es prem el botó desitjat de l'escena.

Per a la tria de la funció que s'ha d'executar amb la selecció del botó s'ha de triar en primer lloc el Game Object que porta associat l'script on hi ha la funció que es vol executar. A continuació al desplegable de la dreta es selecciona la funció. Cal remarcar que aquesta funció ha d'estar definida com a funció pública per tal de poder-hi accedir des de fora de l'script. Finalment, si aquesta funció porta inclosa alguna variable, aquesta s'introdueix en el requadre que hi ha a sota del

desplegable.

Per a una millor comprensió es recomana observar la següent imatge. En ella hi ha un exemple en que es fa ús de la funció On Click que porten incorporada tots els botons.

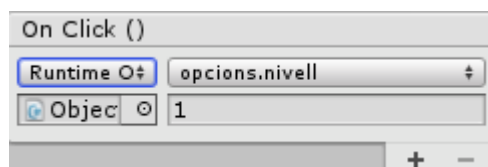


Fig. 9.12. Funció On Click() dels elements botó

Per facilitar l'execució de la tasca, als diferents rangs de dificultat se'ls ha fet correspondre un valor enter, quedant expressats de la següent manera:

Botó	Baix	Mig	Alt
Enter	1	2	3

I els idiomes s'expressen amb un string "cat" i "cast".

La funció On Click() dels botons Baix, Mig i Alt crida la variable entera nivell, que es troba a l'script opcions associat a l'ObjecteBuit, fent-la correspondre amb el valor de la variable d'entrada indicat al requadre amb els valors de 1, 2 i 3 respectivament.

La funció On Click() dels botons Català i Castellà crida la variable idioma, que és un string que indica si l'idioma seleccionat és el català "cat" o el castellà "cast". Aquesta variable es troba a l'script opcions associat a l'ObjecteBuit i es fa correspondre amb el valor de la variable d'entrada indicada al requadre amb els valors de "cat" i "cast" en funció del botó que es premi.

#### 9.2.2.5. Botó torna

A l'extrem inferior dret de l'escena s'observa una imatge d'una fletxa simbolitzant una tecla de retornar a la pantalla anterior. La funció d'aquest botó, per tant, és fer la transició entre la pantalla d'opcions i la pantalla principal un cop s'ha acabat de definir les opcions desitjades.

- Components i assets

L'estructura i els components del botó és la mateixa que s'ha explicat a l'ap. 9.2.2.4. En

aquest cas s'ha fet ús del component Image, al que se li ha afegit un asset en forma d'imatge que es correspon amb la imatge que es veu a l'escena, la fletxa.

- Scripts

També es veu modificada la part de la funció On Click(), ja que ara el que fa és crida a funció torna, ubicada a l'script opcions associat a l'objecte buit. Aquesta funció s'encarrega de fer la transició entre la pantalla actual i la pantalla\_principal. La funció crida el mètode LoadLevel de la classe Application tal i com es mostra en l'ap. 9.1.2.4.

#### 9.2.2.6. ObjecteBuit

Els objectes buits estan presents en l'escena però no de manera visible, per aquest motiu no es podrà interaccionar mai amb ells a través del ratolí. Tot i així, són molt importants en el desenvolupament de l'aplicació ja que en ells normalment s'hi afegeix l'script que dóna cos a l'escena i que conté informació clau per al seu correcte funcionament.

- Components i assets

En esser un objecte buit, aquest únicament conté el component Transform, que permet modificar convenientment la seva posició.

- Scripts

En aquest cas l'objecte està associat a l'script opcions, que conté les següents funcions:

- Start(): que crida la funció seleccio\_idi i seleccio\_niv per a que les variables nivell i idioma prenguin els valors que se'ls ha assignat per defecte.
- Set\_nivell (int x): crida la funció seleccio\_niv amb un valor d'entrada de x.
- Set\_idioma (string y): crida la funció seleccio\_idi amb un valor d'entrada y.
- Nivell\_Value(): retorna el valor de la variable nivell.
- Idioma\_Value(): retorna el valor de la variable idioma.
- Torna(): carrega l'escena principal.
- Seleccio\_niv (int x): en primer lloc posa a blanc el fons de tots els botons que fan referència al nivell i a continuació posa a groc el botó que s'hagi premut per última vegada.
- Seleccio\_idi (string y): posa en groc la última selecció i la manté fins que se'n fa una

altre.

### 9.3. Pantalla\_joc

La pantalla\_joc és la pantalla més complexa del joc ja que incorpora una gran quantitat d'objectes i de scripts. En ella s'hi troben els components que permeten que el pacient, nen o nena, desenvolupin la tasca de segmentació de síl·labes tot visualitzant el nom i la imatge de la paraula i amb la possibilitat d'escoliar-ne un àudio.

En aquesta pantalla és on transcorre l'acció i on realment l'usuari desenvolupa les seves habilitats lingüístiques. L'escena està dissenyada de manera que sigui fàcil d'utilitzar i que tingui un funcionament entenedor.



Fig. 9.13. Vista pantalla joc

#### 9.3.1. Funcionament

El funcionament de la pantalla de joc es basa en la tasca de separació sil·làbica que duen a terme avui en dia a l'hospital per a la rehabilitació logopèdica, la qual forma part d'un conjunt anomenat maleta de jocs. Aquesta tasca avui en dia es fa manualment amb el suport d'imatges en paper, però l'execució de la tasca en suport informàtic és la mateixa.

Actualment la tasca es desenvolupa de la següent manera: En primer lloc el pacient veu la imatge d'un objecte i en llegeix el seu nom i a continuació intenta separar el mot en síl·labes. En cas de no tenir-ho clar, el logopeda llegeix el mot tantes vegades com l'usuari ho requereixi. Un cop té decidida la seva resposta pronuncia el número de síl·labes en què creu que està dividida la paraula.

Amb l'aplicació l'únic que varia són els components i el seu suport físic, per tant, la tasca

anterior es traduiria de la següent manera: En primer lloc el pacient observa la imatge en pantalla, podent llegir el nom a sota d'aquesta. Seguidament, si no està segur de la resposta pot prémer la imatge de l'altaveu, fent que es reproduïxi l'àudio que té enregistrat la lectura del mot. El pacient pot practicar la pronunciació prement el micròfon i dient el número corresponent a les síl·labes que té la paraula; en cas d'encertar sortirà en pantalla una careta somrient i en cas de fallar una careta trista. Finalment, l'usuari prem el botó corresponent al número de síl·labes, apareixent un tick en cas que l'entrada sigui correcta i una creu en cas que sigui incorrecta.

Aquest reconeixement de veu es duu a terme mitjançant la metodologia Dynamic Time Warping, comentada a l'ap. 8.1.3, que consisteix en la comparació de dues llistes de freqüències. En aquest cas les llistes seran:

1. Les freqüències de l'àudio que es capta pel micròfon.
2. Els fitxers model que contenen la llista de freqüències dels números (1,2,3,4,5).

A partir de les llistes es determina la distància que hi ha entre elles per identificar el número que s'ha pronunciat pel micròfon, que es correspon amb la llista que té una distància més petita. Al llarg d'aquest apartat es va descrivint en detall com s'ha desenvolupat aquest procediment.

Les variants de la pantalla d'opcions que afecten al desenvolupament són l'idioma del text de la pregunta, del mot i l'àudio i el nivell, que permet l'aparició de paraules més llargues o amb més síl·labes amb un nivell superior o l'aparició de paraules curtes i de poques síl·labes amb els nivells inferiors.

### **9.3.2. GameObjects**

#### **9.3.2.1. Fons**

L'objecte que fa de fons de pantalla és exactament igual per totes les escenes. En aquest cas tampoc se li ha afegit cap component addicional i per tant, la seva estructura és idèntica a la que s'ha comentat a l'apartat 9.1.2.2.

#### **9.3.2.2. Main Camera (càmera principal)**

Aquest objecte està comentat a l'apartat 9.1.2.1, però en aquest cas se li s'ha afegit un component addicional.

- Components i assets

Es tracta d'un component AudioSource, que conté molts paràmetres com ara AudioClip,



Output, Mute i Loop i serveix per a reproduir àudios. Aquest component s'ha emprat en el procediment d'escriptura dels fitxers de text que fan de model. Per a dur a terme aquest procés es va reproduir uns àudios enregistrats prèviament que consistien en la pronunciació dels números enters de l'1 al 5 en català i en castellà. Cada vegada que s'havia de generar un fitxer de text es canviava l'asset de l'AudioClip, que correspon al fitxer d'àudio extern, col·locant-hi el desitjat.

- Scripts

Per a dur a terme aquest procediment es va associar l'script crear\_txt a l'objecte Main Camera, però un cop finalitzat es va treure l'script de l'objecte ja que sinó el procediment es duria a terme cada vegada que s'executés l'aplicació i impediria el seu correcte funcionament.

L'script consisteix en la lectura de les freqüències de l'àudio que es reproduïx, el seu emmagatzematge en una llista i el seu posterior traspàs a un fitxer de text.

Per a la lectura de les freqüències d'un àudio es duu a terme el següent procediment:

- **Reproducció de l'àudio** que hi ha dins del component AudioSource de l'objecte Main Camera. Aquesta reproducció es troba dins la funció Start(), permetent la seva execució cada vegada que s'inicia el programa.

```
b = GameObject.Find("Main Camera").GetComponent<AudioSource>();  
  
b.Play();
```

Aquest pas és imprescindible ja que el format del que retorna aquesta funció fa possible el càlcul de les freqüències.

- **Obtenció de les freqüències de l'àudio.** Per a aconseguir-ho s'empra la funció pública GetSpectrumData(llista de floats[], canal d'entrada, FFTWindow window) [11], [12], que un cop es reproduïx l'àudio, el converteix en una llista de períodes que es guarden dins la llista de floats[] que se li dona a l'entrada. Per a convertir-los en freqüències només cal fer la inversa.

```
b.GetSpectrumData(samples, 0, fftwindow);
```

Després de moltes proves, s'ha aconseguit determinar que ha de transcórrer un temps des de la reproducció de l'àudio fins a la seva conversió en freqüències, ja que la conversió és molt ràpida i en cas d'executar-se alhora només capta silencis i valors molt grans de freqüències. Per aquest motiu s'ha incorporat un timer dins de la funció Update() i s'ha definit que la funció GetSpectrumData s'executi quan ha transcorregut 1 segon des de la reproducció.



- **Emmagatzematge en fitxers de text.** Existeixen comandes simples per a crear fitxers de text a partir de la plataforma Unity i per a escriure-hi. Aquestes són:

```
var sr = File.CreateText("uno.txt");

sr.WriteLine((1 / samples[i]).ToString());

sr.Close();
```

La comanda del centre s'executa tantes vegades com nombres hi ha a la llista samples per tal de emmagatzemar a cada línia del fitxer un valor de freqüència.

### 9.3.2.3. Pregunta i nom

L'objecte que mostra el text amb la pregunta "Quin número de síl·labes té la paraula?" fa de títol de l'escena i indica quin és l'objectiu del joc i la tasca que s'ha de dur a terme. Hi ha un altre objecte de tipus text que introdueix el nom de la paraula que s'està visualitzant, permetent així que el pacient també desenvolupi les habilitats de lectura i obtingui informació gràfica, escrita i d'àudio.

Per a satisfer la curiositat del lector, en cas que s'ho estigues preguntant, la formulació de la pregunta que introdueix el joc d'aquesta manera enlloc de "Quantes síl·labes té la paraula?" té els seus motius. Està formulada així perquè tant en català com en castellà la resposta correcta a la segona pregunta "Quantes síl·labes té la paraula?" seria en femení, ja que la paraula síl·laba és femenina, però aquest fet dificultava el reconeixement de veu, motiu pel qual s'ha optat per la pregunta "Quin és el número de síl·labes que té la paraula?".

- Components i assets

Els textos estan formats per un objecte del tipus 3D Text, que com s'ha vist anteriorment conté els components: Transform, Mesh Renderer i Text Mesh. Aquest últim és el més important ja que és en el que s'introdueix el fragment de text que es vol visualitzar i en el que es trien aspectes bàsics del text, com ara la mida, el tipus de lletra i el color.

- Scripts

Aquests objectes no tenen associat cap script ni component adicional com un Box Collider perquè no han de comportar-se de cap manera, són figures per ser llegides i donar informació a l'usuari.

### 9.3.2.4. Números (1,2,3,4,5)

A la part inferior de l'escena s'observen els números de l'1 al 5. Aquests fan de botó i estan

implementats de la mateixa manera que els botons de jugar, opcions, resultats i sortir de la pantalla principal (ap.9.1.2.4). Bàsicament són objectes en 3D del tipus 3D Text als que se'ls ha afegit un component Box Collider per poder interaccionar amb el ratolí.



---

Fig. 9.14. Números amb el 3 seleccionat

- Components i assets

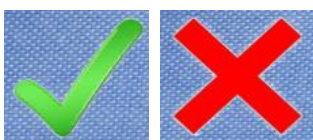
Veure l'ap. 9.1.2.4 ja que tenen exactament la mateixa estructura.

- Scripts

Els números porten associat l'script Botonumero. Aquest conté les funcions OnMouseEnter() i OnMouseExit() comentades anteriorment, que porten comandes per posar el números a groc quan el ratolí els passa per sobre i posar-los en blanc quan surt de sobre a no ser que s'hagi premut el botó.

La funció OnMouseDown() s'executa quan es prem el botó. Aquesta funció en crida moltes altres que es troben ubicades en altres scripts, principalment l'script de joc, que es comentarà més endavant. En essència el que fa aquesta funció és verificar si el botó que s'ha premut correspon al nombre de síl·labes que té la paraula. Per a fer-ho s'ha assignat una variable x a cadascun dels botons fent que prengui el valor del botó i que compari aquest valor amb el nombre de síl·labes.

En cas que el botó sigui el correcte la funció assigna al component Image de l'objecte Correcte/Incorrecte la imatge d'un tick verd i en cas de que sigui incorrecte hi assigna una creu vermella.



---

Fig. 9.15. Imatge tick i creu

Aquesta funció també crida les funcions `get_encertades`, `get_fallades` i `get_puntuacio` que s'encarreguen de portar el recompte de paraules que s'han encertat, fallat i de la puntuació total.

A més a més, la funció porta incloses comandes que fan que l'objecte es quedi de color groc un cop s'ha premut però que torni al seu estat natural cada vegada que es visualitza una nova imatge.

#### 9.3.2.5. Canvas-imatge

Per fer memòria, un canvas és un Game Object amb un component canvas que té la funció d'espai abstracte on es col·loquen tots els elements de la interfície d'usuari (UI).

El canvas-imatge conté elements de la interfície d'usuari com un panell amb una imatge, el botó següent, el botó altaveu i dos objectes buits. En primer lloc es comentaran els objectes buits i els seus scripts ja que són els més importants d'aquesta escena i els que contenen la informació clau i les funcions que controlen el transcurs d'esdeveniments.

#### 9.3.2.6. Objectebuit1

Aquest objecte no està present a l'escena de manera visual però és essencial per al desenvolupament de la tasca ja que controla l'altaveu i determina el seu funcionament.

La funció d'aquest objecte és la de reproduir el l'àudio que conté la pronunciació de la paraula que s'està visualitzant en el moment en que es premi el botó de l'altaveu. L'usuari podrà prémer tantes vegades com vulgui l'element altaveu per tal de poder escoltar la paraula.

- Components i assets

A l'objecte, que tenia només com a component per defecte el Transform, se li ha afegit un component AudioSource, que com s'ha vist s'utilitza per a la reproducció d'àudios.

- Scripts

S'ha associat a aquest objecte l'script altaveu, que té la funció de reproduir el contingut del paràmetre AudioClip del component AudioSource. Per a reproduir el so fa servir la comanda:

```
GetComponent<AudioSource>().Play();
```

Que està inclosa dins de la funció Reproduir() que és la que es defineix dins de la funció OnClick del botó altaveu.

### 9.3.2.7. Altaveu i Següent

Altaveu i següent són dos GameObjects del tipus UI Button que estan situats a la part dreta de l'escena. El primer té la funció de reproduir l'àudio de la paraula quan sigui premut i el segon permet fer la transició entre imatges.



Fig. 9.16. Imatge element altaveu i fletxa

- Components i assets

Els components d'un UI Button s'han comentat a l'apartat 9.2.2.4.

- Scripts

Seguint les instruccions donades en l'ap. 9.2.2.4, s'introdueix dins de la funció On Click() del botó Altaveu la funció Reproduir() associada a l'objectebuit1 que s'ha mencionat en l'apartat anterior.

A la funció On Click() de botó Següent s'introdueix la funció canvia associada a l'objectebuit2 que es comentarà seguidament. Aquesta funció permet fer la transició entre les visualitzacions del conjunt de les imatges que formen part del joc.

### 9.3.2.8. Objectebuit2

Aquest objecte conté la informació necessària per al desenvolupament del joc ja que s'encarrega de lectura de la base de dades, de la seva interpretació, de la transició entre paraules de joc i del càlcul de la puntuació.

- Components i assets

L'objectebuit2 està format pel component Transform, que venia per defecte.

- Scripts

Aquest objecte té associat un script que és un dels més importants i complexos de tota

l'aplicació ja que és el que dona cos a l'escena.

Al començament, com en tots els scripts es troba la definició de les variables que s'empraran en el codi i, en aquest cas, també es troba la definició d'una nova estructura anomenada pack, que conté una paraula, una imatge, un so i un enter indicatiu del nombre de síl·labes. Aquest paquet el que fa és convertir la informació en forma de strings de l'entrada en informació d'altres tipus.

A continuació se'n pot veure la definició:

```
public string paraula;
public Sprite foto;
public AudioClip so;
public int silabes;
public pack(string n, string f, string s, string sil)
{
    paraula = n;
    foto = Resources.Load<Sprite>(f);
    so = Resources.Load(s) as AudioClip;
    silabes = int.Parse(sil);
}
```

Seguidament es defineix la classe joc, derivada de la classe MonoBehaviour, que és la que s'encarrega de desenvolupar la comparació entre les dues llistes de freqüències, que s'han comentat a l'inici de l'apartat, per executar el reconeixement de veu amb la metodologia DTW. La classe està formada per vàries funcions:

- Main(): És una funció pròpia del programa i en aquest cas s'encarrega de llegir el document XML que ha definit anteriorment el logopeda i crear un paquet (definit a dalt) a partir de cadascun dels elements que hi ha al fitxer i dels seus atributs.
- Start(): Aquesta funció s'executa a l'inici de la visualització de l'escena i en aquest cas fa que es generi una llista, anomenada total, de 8 nombres aleatoris que vagin des del 0 fins al nombre de elements de l'xml. Es crida la funció visualitza per al primer element de la llista.
- Canvia(): Aquesta funció posa de color blanc els números de la pantalla i, si encara queden elements a la llista de nombres aleatoris, crida la funció visualitza del següent de la llista. També posa en blanc l'objecte Correcte/Incorrecte quan es canvia de paraula i, en cas d'haver finalitzat la llista total, fa la transició cap a la pantalla de resultats.
- Visualitza(): Primerament la funció inhabilita l'objecte fletxa per a que els usuaris no puguin eludir cap pantalla. A continuació, introdueix la imatge que pertorqui a la paraula que es visualitza dins del component sprite de l'objecte Imatge que es

comentarà més tard. Mitjançant el paquet definit anteriorment amb les dades de la base de dades també s'obté el so que li correspon a la paraula que s'està treballant i l'adjudica al component AudioSource de l'objectebuit1. Finalment, mitjançant el paquet troba el nom de la paraula i la mostra a la pantalla fent-la correspondre amb el component TextMesh de l'objecte Nom.

- GetSil(): La funció activa el funcionament de l'objecte fletxa un cop l'usuari ha premut un dels números. D'aquesta manera es garanteix que no passin de paraula sense haver contestat l'anterior. La funció retorna el valor numèric del nombre de síl·labes que té la paraula amb la que s'està treballant.
- Get\_apretat() i clickat(): sumen 1 a una variable. Aquestes funcions es criden a l'script de botonumero per saber si l'usuari a introduït la seva resposta prement un dels números.
- Get\_encertades(): suma 1 a la variable encertades. Aquesta funció només es crida quan el nombre de síl·labes coincideix amb el número del botó que s'ha premut.
- Get\_fallades(): suma 1 a la variable fallades i es crida només quan el nombre de síl·labes no coincideix amb la resposta de l'usuari.
- Get\_puntuacio(): retorna la puntuacio acumulada, calculada com  $\text{encertades} * 5 - \text{fallades}$ ; ja que se sumen 5 punts per cada resposta encertada i se'n resta 1 per cada resposta fallada.

### 9.3.2.9. Imatge

Aquest objecte permet la visualització d'una imatge que fa referència a la paraula que s'està treballant.

- Components i assets

L'objecte imatge consta de components per defecte com el Rect Transform, el Canvas Renderer o el Image. Aquest últim té un paràmetre anomenat Source Image que és on s'hi introdueix l'sprite que es vol visualitzar. El canvi d'sprite es fa via codi dins de la funció visualitza de l'script joc de l'objectebuit2.

- Scripts

No té associat cap script ja que només serveix d'eina de visualització i segueix les ordres de l'script joc.

#### 9.3.2.10. Canvas-tick i creu

El canvas imatges conté elements de la interfície d'usuari com un panell amb una imatge i un botó Micro. Veure la definició de canvas a l'ap. 9.3.2.5.

#### 9.3.2.11. Correcte/Incorrecte

Aquest objecte permet la visualització d'una imatge en forma de tick quan l'usuari ha encertat el nombre de síl·labes de la paraula i una imatge en forma de creu quan l'usuari ha fallat el nombre de síl·labes. El seu funcionament i estructura és molt similar al de l'objecte Imatge (ap.9.3.2.9).

- Components i assets

L'objecte imatge consta de components per defecte com el Rect Transform, el Canvas Renderer o el Image. Aquest últim té un paràmetre anomenat Source Image que és on s'hi introdueix l'sprite que es vol visualitzar. El canvi d'sprite es fa via codi dins de la funció OnMouseDown de l'script botonumero dels objectes números.

- Scripts

No té associat cap script ja que només serveix d'eina de visualització i segueix les ordres de l'script joc.

#### 9.3.2.12. Micro

L'objecte Micro és de tipus UI Image i és l'encarregat de fer el reconeixement de veu. El seu funcionament és similar al d'un botó i consta de diversos passos:

- En el moment en què l'usuari prem la imatge, s'activa el micròfon i el que s'ha dit en els moments posteriors a la seva activació és enregistrat.
- A partir de l'àudio registrat es segueix el mateix procediment de conversió a freqüències que en l'script crear\_txt.
- Un cop es té la llista de freqüències, aquestes s'enregistren en un altre fitxer txt, per si es volgués fer un altre tractament amb elles, i es comparen els valors amb els 5 fitxers model.
- Mitjançant la metodologia de reconeixement de veu Dynamic time warping, explicada a l'ap. 8.1.3, es troba la diferència que hi ha entre el fitxer obtingut pel micròfon i cadascun dels fitxers models.

- Es busca la distància mínima que teòricament correspondrà amb el número que s'ha entrat pel micròfon.




---

Fig. 9.17. Imatge element Micro

- Components i assets

L'objecte micro conté els components per defecte d'un objecte d'aquest tipus: Rect Transform, Canvas Renderer i Image. Aquest últim té assignat l'asset corresponent a una imatge anomenada micro dins del paràmetre Source Image.

A més a més s'han inclòs dos components més, que són un Box Collider per a que l'objecte pugui interaccionar amb el ratolí i pugui ser premut i un component Audio Source, que serveix per captar l'àudio pel micròfon.

- Scripts

El micro porta associat un script anomenat comparador, que és l'encarregat de fer el reconeixement de veu mitjanant la metodologia DTW. Aquest script conté diverses funcions:

- OnMouseDown(): un cop es clicka amb el ratolí sobre la imatge, s'assigna al component AudioSource del Micro el dispositiu micròfon i, per tant, capta el so que rep d'aquest dispositiu i el reproduceix.
- Update(): es duu a terme el procediment d'obtenció de les freqüències de la mateixa manera que en l'ap.9.3.2.2 i s'emmagatzemen les dades en un fitxer de text anomenat "audio\_micro". Quan han passat 2 segons es tanca el micròfon.
- DTWDistance(s,t): és la traducció del codi trobat en la referència [4] facilitada per la tutora del treball. Aquesta funció consisteix en determinar quina distància hi ha entre els valors de les dues llistes d'entrada, essent una la llista de freqüències model i l'altre la llista de freqüències captada del micròfon. Com menor sigui la distància DTW més similars són dos àudios.



- Compara(s,a): és una funció que crida la funció Load per a cadascun dels valors que prengui el float d'entrada a i crida la funció DTWDistance (s, freq), essent s la llista de freqüències del micròfon i freq la llista de freqüències de l'àudio model llegit amb la funció Load.
- Load(filename): funció que llegeix el fitxer de text que es correspongui amb el nom d'entrada i retorna la llista dels valors que hi té emmagatzemats.
- Tria(s): és una funció que crida la funció compara per als 5 fitxers model, enregistra els valors de les distàncies en una llista, en troba el mínim i retorna el nombre que s'ha interpretat. Si el nombre es correspon amb el nombre de síl·labes es canvia la imatge del micròfon per una careta somrient i sinó es canvia per una careta trista.

Cal mencionar que aquest script es pot optimitzar molt ja que el temps d'execució és molt elevat i també perquè ha donat molts problemes en el transcurs del seu desenvolupament. Es creu que es podrien provar altres maneres de fer el reconeixement de veu, com per exemple intentar fer ús del reconeixement de veu ja desenvolupat d'altres plataformes o sistemes informàtics de manera externa i intentar incorporar els resultats a l'aplicació.

## 9.4. Pantalla\_resultats

La pantalla resultats mostra la puntuació obtinguda, així com el nombre d'encerts i fallades. D'aquí el logopeda pot extreure les seves pròpies conclusions i observar l'evolució que està tenint el pacient.

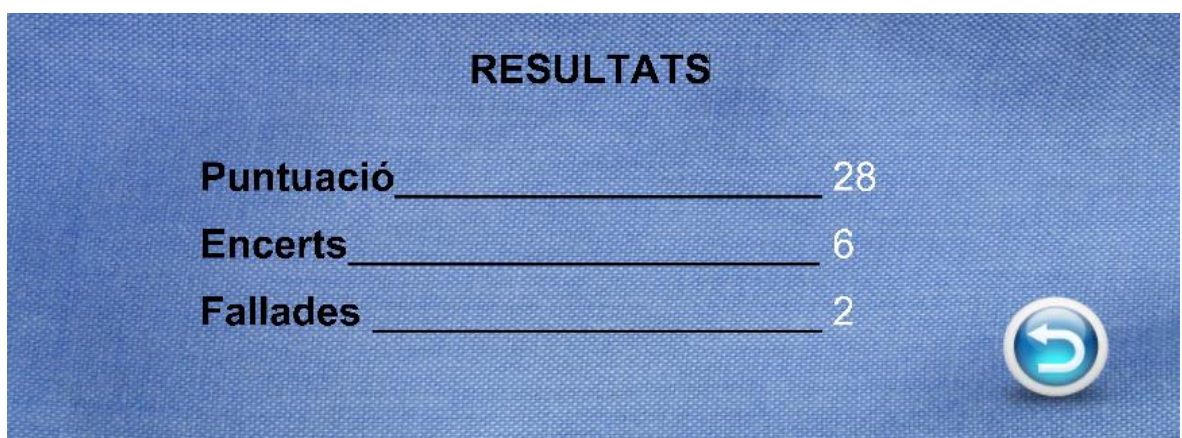


Fig. 9.18. Vista pantalla resultats

### 9.4.1. Funcionament

El funcionament d'aquesta pantalla és molt senzill ja que només s'usa per a la visualització dels resultats, no hi ha cap altra interacció ni per part de l'usuari ni per part del logopeda. Els resultats es mostren directament en pantalla quan s'acaben les paraules a treballar en la pantalla\_joc.

Existeix un botó en aquesta pantalla, el botó return que permet la transició de la pantalla\_resultats a la pantalla\_principal.

Cal remarcar que es pot accedir a la visualització de resultats tantes vegades com es desitgi prement el botó resultats de la pantalla principal ja que aquests només s'esborren en el moment en que s'inicia un altre joc.

### 9.4.2. GameObjects

#### 9.4.2.1. Main Camera (càmera principal)

Aquest objecte està comentat a l'apartat 9.1.2.1 i els seus components són els mateixos ja que no s'ha incorporat cap component addicional en la càmera d'aquesta escena. També porta incorporat l'script que permet l'adaptació del contingut de l'aplicació a la mida de pantalla del dispositiu que es fa servir, anomenada AspectUtiliy.

#### 9.4.2.2. Fons

L'objecte que fa de fons de pantalla és exactament igual per totes les escenes. En aquest cas tampoc se li ha afegit cap component addicional i per tant, la seva estructura és idèntica a la que s'ha comentat a l'apartat 9.1.2.2.

#### 9.4.2.3. Resultats, Puntuació, Fallades i Encerts

L'objecte resultats mostra el text "RESULTATS" i fa de títol de l'escena, introduint el contingut d'aquesta, mentre que els objectes puntuació, fallades i encerts introdueixen els diferents resultats que es mostren en pantalla.

- Components i assets

Els quatre textos estan formats per un objecte del tipus 3D Text, que com s'ha vist anteriorment conté els components: Transform, Mesh Renderer i Text Mesh. Aquest últim és el més important ja que és en el que s'introdueix el fragment de text que es vol visualitzar i en el que es trien aspectes bàsics del text, com ara la mida, el tipus de lletra i el color.

- Scripts

Els objectes Puntuació, Fallades i Encerts no tenen cap script associat, però en canvi l'objecte Resultats sí que en té un. Aquest és l'script resultats, que permet la visualització dels resultats.

La funció més important és la funció Start() ja que en ella es troben els components Text Mesh dels objectes Puntuació, Fallades i Resultats als quals se'ls assigna el valor de les variables de l'script joc de la pantalla\_joc de la següent manera:

```
d = GameObject.Find("Valor_puntuacio").GetComponent<TextMesh>();  
d.text = Convert.ToString(joc.encertades*5-joc.fallades);
```

L'assignació mostrada és la del Valor\_puntuació però les altres dues seguirien la mateixa estructura.

També hi ha la funció torna() que permet la transició a la pantalla principal.

#### 9.4.2.4. Valor\_puntuacio, Valor\_encerts i Valor\_fallades

Aquests elements estan afegits per mostrar el resultat obtingut. Els valors per defecte que tenen els objectes és 0, però quan s'arriba a la pantalla resultats després d'haver jugat en la seva inicialització s'actualitzen els valors.

- Components i assets

Els objectes són de tipus 3D Text i tenen els components per defecte: Transform, Mesh Renderer i Text Mesh. Aquest últim és molt important ja que en aquesta pantalla ha d'anar canviant el seu contingut en funció del valor que s'obtingui del joc.

- Scripts

No tenen cap script associat.

#### 9.4.2.5. Botó torna

Aquest botó serveix per tornar a la pantalla principal un cop ja s'han visualitzat els resultats i s'han tret les conclusions pertinents.

- Components i assets

Els objectes del tipus UI Button tenen, com s'ha vist, els components Rect Transform, Canvas Renderer, Image i Button per defecte. En aquest cas s'ha afegit l'asset de la imatge anomenada torna (de la carpeta Assets) al component Target Graphic, de manera que el

botó pren la forma de la imatge.

- Scripts

A la funció On Click() del component Button se li ha assignat la funció torna associada a l'objecte Resultats. D'aquesta manera en el moment en que es prem el botó es produeix la transició a la pantalla principal.

## 10. Treball futur

En aquest apartat s'introduiran possibles millores o properes etapes que hauria de tenir el projecte en cas que es disposés de més temps per a elaborar-lo.

El funcionament de l'aplicació en sí és correcte, tot i que els resultats del reconeixement de veu no són 100% fiables ja que no sempre interpreta bé l'àudio que se li entra. Aquest fet pot ser degut a múltiples causes, com ara el soroll que pot tenir el senyal de veu captat pel micròfon o bé els possibles errors en els fitxers de freqüències que fan de model per a la comparació.

En un futur es podria treballar amb els fitxers de freqüències i estudiar-ne la variabilitat, enregistrant-los diverses vegades comprovant com de diferents són els seus valors per cadascun dels enregistraments. D'aquesta manera es determinaria si realment es pot captar una llista de freqüències que segueixi un patró específic i diferenciador per cada número que es pronuncia.

Per acabar de millorar el reconeixement de veu l'autora del treball es podria posar en contacte amb membres del projecte *Comunica*, de la *Universidad de Zaragoza*, autors de l'aplicació *Vocaliza* (ap. 4.2) per a demanar consell i orientació sobre el tema i les possibles millores que s'hi podrien fer.

Finalment, per a millorar l'aplicació es podrien prendre altres dades o mesures per a la visualització dels resultats, com ara el temps de resposta, o també es podria crear una base de dades que enregistrés els resultats i, per exemple, les paraules més problemàtiques per al pacient. Així es facilitaria més informació al logopeda i aquest podria dur un seguiment més acurat de l'evolució del pacient.

## 11. Planificació temporal

En un projecte és molt important la planificació temporal ja que d'ella depèn que el projecte es dugui a terme dins del temps establert del projecte. Així doncs, per a elaborar la planificació temporal del projecte es va tenir en compte totes les tasques que s'havien de fer:

1. **Contacte amb la tutora del treball i definició dels objectius:** Explicació del projecte, de l'aplicació a implementar i dels requeriments.
2. **Estudi de mercat:** es fa una cerca de les aplicacions que existeixen avui en dia que tenen semblances amb la que es vol implementar. Se n'extreuen idees importants i es pensa quina es podria incorporar al treball.
3. **Anàlisi dels usuaris:** anàlisi de les persones que estaran en contacte amb l'aplicació i de les seves necessitats.
4. **Definició de la tasca i de com es desenvoluparà:** determinació dels aspectes més importants que tindrà el joc i del punt de partida.
5. **Introducció de l'entorn de Unity:** descàrrega de la plataforma i familiarització amb el programa per obtenir un cert domini per poder començar a treballar-hi.
6. **Introducció al llenguatge de programació C#:** Aquest és un llenguatge de programació desconegut i cal un temps d'autoaprenentatge mitjançant l'ús de tutorials trobats a la xarxa [1] i [2].
7. **Estudi de possibles sistemes de reconeixement de veu:** per a dur a terme la tasca cal fer un estudi dels diferents sistemes de reconeixement que existeixen avui en dia per triar quin seria més factible en aquest cas.
8. **Realització del fitxer XML:** descarrega del programa XML Notepad, familiarització amb aquest tipus d'arxiu i escriptura del fitxer.
9. **Implementació de l'aplicació i cerca d'informació per a fer-ho:** desenvolupament de l'aplicació amb el suport de la documentació de Unity.
10. **Detecció i correcció d'errors:** cal comprovar tots els codis i els seu funcionament. Per això cal testear-los, trobar els punts dèbils i corregir-los.
11. **Redacció de la memòria:** redactat de la memòria, incloent tots els passos que s'han dut a terme per al desenvolupament de l'aplicació.

A continuació s'ha fet una estimació del temps que s'ha dedicat a cadascuna de les parts:

Tasca	Contacte amb la tutora i definició dels objectius	Estudi del mercat	Anàlisi dels usuaris	Definició de la tasca i de com es desenvoluparà	Introducció de l'entorn de Unity	Introducció al llenguatge de programació C#
Estimació de temps dedicat (hores)	3	8	6	12	25	35

Tasca	Estudi de sistemes de reconeixement de veu	Realització fitxer XML	Implementació de l'aplicació	Detecció i correcció d'errors	Redacció de la memòria
Estimació de temps dedicat	25	6	120	30	60

En total s'estima que s'ha invertit al voltant de 330 hores en un període de 4 mesos aproximadament. A continuació es mostra la planificació en un diagrama de Gantt. S'ha distingit les etapes en tres grups, que es mostren en tres colors diferents: en groc la introducció a la tasca, en color violeta la familiarització amb el contingut i l'entorn de treball i en taronja el desenvolupament de l'aplicació.

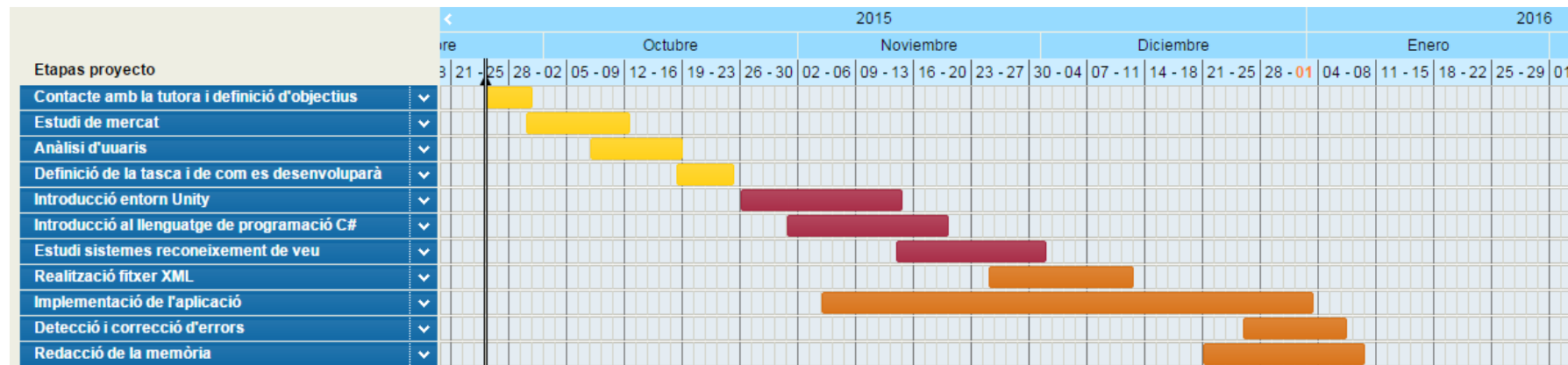


Fig. 11.19. Diagrama de Gantt



## 12. Costos

Per a fer el còmput dels costos d'aquest projecte s'ha tingut en compte tant el temps dedicat com el material emprat. Per tant, s'ha subdividit el conjunt de costos en dos tipus: Costos associats als treballadors i costos associats al material.

- Costos dels treballadors o de recursos humans:

Aquest projecte ha estat desenvolupat per una única treballadora exercint el rol de graduada en Enginyeria en Tecnologies Industrials. La mitjana salarial d'un enginyer acabat de titular està al voltant dels 1850€/mes i la mitjana d'hores treballades al més al voltant de 134, la qual cosa fa que la mitjana salarial per hora sigui d'uns 14 €/h.

Basant-se en les hores dedicades a cada etapa del procés es fa el següent càlcul de costos:

ETAPA	Temps (h)	Cost (€)
<b>Contacte amb la tutora i definició dels objectius</b>	3	42
<b>Estudi de mercat</b>	6	84
<b>Anàlisi dels usuaris</b>	8	112
<b>Definició de la tasca i de com es desenvoluparà</b>	12	168
<b>Introducció a l'entorn de Unity</b>	25	350
<b>Introducció al llenguatge de programació C#</b>	35	490
<b>Estudi de sistemes de reconeixement de veu</b>	25	350
<b>Realització del fitxer XML</b>	6	84
<b>Implementació de l'aplicació</b>	120	1680
<b>Detecció i correcció d'errors</b>	30	420
<b>Redacció de la memòria</b>	60	840
<b>Total (€)</b>		<b>4270</b>

- Costos associats al material:

Per al desenvolupament i implementació de l'aplicació ha estat necessari en tot moment l'ús d'un equip informàtic, però els costos d'aquest equip no es tindran en compte ja que ja es disposava d'aquest element, ni tampoc es tindran en compte els costos de desgast de material ja que es consideren negligibles. El treball s'ha realitzat amb un ordinador Acer amb sistema operatiu Windows 7.

El programari que s'ha emprat ha estat totalment gratuït, tant la versió de Unity que s'ha fet servir com el programa XML Notepad que s'ha descarregat per al processament de l'arxiu XML.

El que sí que s'ha de tenir en compte és el cost d'Internet i el cost del consum elèctric, ja que tota la informació s'ha extret de la xarxa i, a més, la plataforma Unity requereix Internet per funcionar. És difícil estimar el consum d'electricitat de la xarxa de l'ordinador i del temps que hi ha estat connectat, per tant només es farà una estimació del cost d'Internet.

Cost	Preu (€/mes)	Temps (mesos)	Cost (€)
Internet	45	4	180

El total de costos sumen una xifra de 4450€, per tant aquest és el cost que ha tingut l'elaboració d'aquest projecte.

## 13. Impacte en l'entorn

L'impacte més evident que tindrà l'elaboració d'aquest projecte en l'entorn és sobretot de caràcter social. L'aplicació que s'ha desenvolupat és una eina de suport als logopedes i, per tant, tindrà un impacte molt positiu entre els professionals del sector. A més, l'objectiu de l'aplicació és el d'ajudar a nens i nenes en el seu entrenament de les habilitats lingüístiques i, en definitiva, en la seva rehabilitació logopèdica. Per aquest motiu, s'espera que l'app. tingui una gran acceptació i que contribueixi en el desenvolupament de les capacitats lingüístiques dels pacients de l'àrea infantil de l'hospital.

D'altra banda, com s'ha comentat anteriorment, el desenvolupament de l'aplicació ha requerit l'ús de dispositius que necessiten energia elèctrica per al seu funcionament. Aquest precisament seria un impacte en el medi, encara que la quantitat consumida és tan petita que es pot considerar nul·la.

Finalment, també cal tenir present que s'està potenciant l'ús d'eines informàtiques que comporten despesa d'energia elèctrica en l'ordinador. Es considera que la despesa que podria suposar l'ús d'aquesta aplicació a l'Hospital Sant Joan de Déu, dins del conjunt de la societat no tindria efectes significatius a nivell energètic.

## Conclusions

El resultat d'aquest treball ha estat el desenvolupament d'una aplicació en la que es poden treballar tant habilitats semàntiques com habilitats de la parla i que està destinada a donar suport a la feina dels logopedes en la rehabilitació lingüística d'infants de l'Hospital Sant Joan de Déu de Barcelona. L'objectiu principal que era la implementació d'una de les tasques que duen a terme a l'hospital avui en dia de manera manual ha tingut resultats satisfactoris ja que s'ha aconseguit crear una estructura de joc fàcil d'utilitzar per infants i fàcil de manejar per als logopedes.

L'eina dissenyada compleix amb tots els requeriments que s'havien imposat al principi del treball. L'aplicació és dinàmica ja que permet incorporar informació i modificar-ne el contingut, té paràmetres variables que permeten adaptar el joc a les necessitats de l'usuari i facilita la visualització de resultats, fet que fa que el logopeda pugui portar un control de l'evolució del pacient. En aquesta visualització es pot observar el nombre d'encerts i fallades i la puntuació total, encara que hi ha altres mesures que no s'han pres però que es podrien considerar, com per exemple el temps que es triga en respondre. A més, conté una base de dades en català i una en castellà permetent treballar amb tots dos idiomes.

Fent memòria, un dels objectius d'aquest projecte també era la verificació de si era possible implementar tecnologia de reconeixement de veu en la plataforma Unity i, després de l'elaboració d'aquesta aplicació, es pot concloure que avui en dia les funcionalitats que hi ha a Unity no permeten un reconeixement de veu fiable en idiomes que no siguin l'anglès. Per aquest motiu ha estat necessari implementar funcions com l'algorisme DTW.

A més a més, cal comentar que la tecnologia pròpia de Unity referent a reconeixement de veu s'usa molt ocasionalment, tenint inclús funcions obsoletes i poca informació en la documentació. Per a futurs projectes es podia pensar en la utilització de tecnologies de reconeixement que ja funcionen i estan en ús en el mercat, més complexes i treballades, i intentar acoblar el seu funcionament amb el motor de videojocs de Unity.

L'implementació d'aquesta aplicació ha suposat un repte ja que no es coneixia l'entorn de treball ni el llenguatge de programació. No obstant, ha servit de preparació per a la vida professional ja que no només se n'obtenen coneixements informàtics sinó també ha servit d'aprenentatge de com enfrontar-se a un projecte de temàtica poc coneguda i portar-lo endavant per tu mateix.

## Agraïments

Agrair a la Núria Pla i al Marc Vigo l'orientació i ajuda en el transcurs del treball, així com l'oportunitat de participar en un projecte de col·laboració amb un hospital d'aquesta envergadura.

Agraeixo també al departament de Ciències de la computació que m'hagin facilitat els treballs del Carles Regué i l'Albert Sauri ja que han estat molt útils i han servit de referència per a l'estructura del treball.

Finalment, agraeixo a la meva parella i amics de la universitat les seves aportacions i col·laboració en la resolució de dubtes de caire informàtic i als familiars i amics el seu suport incondicional.

# Bibliografia

## Referències bibliogràfiques

- [1] **Technologies, Unity.** Learn Tutorials: *Creating a breakout game for beginners.* [https://unity3d.com/es/learn/tutorials/modules/beginner/live-training-archive/creating-a-breakout-game?playlist=17219, 8 d'octubre del 2015]
- [2] **Technologies, Unity.** Learn Tutorials: *Scripting* [https://unity3d.com/es/learn/tutorials/topics/scripting, 12 d'octubre del 2015]
- [3] **Technologies, Unity.** Unity Manual: *UI (Interfaz de usuario)* [https://unity3d.com/es/learn/tutorials/topics/scripting, 4 de novembre del 2015]
- [4] **Wikipedia.** *Dynamic time warping.* [https://en.wikipedia.org/wiki/Dynamic\_time\_warping, 30 d'octubre del 2015]
- [5] **Wikipedia.** Speech recognition. [https://en.wikipedia.org/wiki/Speech\_recognition, 30 d'octubre del 2015]
- [6] **Wikipedia.** Formante. [https://es.wikipedia.org/wiki/Formante, 30 d'octubre del 2015]
- [7] **Wikipedia.** Señal de voz. [https://es.wikipedia.org/wiki/Se%C3%B1al\_de\_voz, 30 d'octubre del 2015]
- [8] **Junta de Andalucía, Averroes.** *Iniciación a la lectoescritura* [http://www.juntadeandalucia.es/averroes/html/adjuntos/2008/04/11/0001/, 12 d'octubre del 2015]
- [9] **Logopeda en Casa.** Speechviewer III. [http://www.logopedaencasa.es/aplicaciones/speechviewer-iii/, 12 d'octubre del 2015]
- [10] **Grupo de las tecnologías de las comunicaciones, Univeridad de zaragoza.** Vocaliza. [http://dihana.cps.unizar.es/~alborada/herramientas.html, 12 d'octubre del 2015]
- [11] **Technologies, Unity.** *GetSpectrumData* [http://docs.unity3d.com/ScriptReference/AudioSource.GetSpectrumData.html, 5 de novembre del 2015]
- [12] **Community Answers, Unity.** *GetOutputData and GetSpectrumData.*

[<http://answers.unity3d.com/questions/157940/getoutputdata-and-getspectrumdata-they-represent-t.html>, 5 de novembre del 2015]

[13] **Johnny Manson, C# Development Tutorial. Voice Recognition.**  
[<https://www.youtube.com/watch?v=AB9lfHDOe5U>, 9 de novembre del 2015]

## Bibliografia complementària

[14] **Mr. Yocyf. Haciendo menus en Unity.**  
[<https://www.youtube.com/watch?v=u2OVSDJwYuw>, 25 d'octubre del 2015]

[15] **Kaapine Tutorials. Using Microphone input in Unity3D.**  
[<http://www.kaappine.fi/tutorials/using-microphone-input-in-unity3d/>, 16 de novembre del 2015]

[16] **Hagamos Videojuegos. Musica y sonidos.**  
[<https://www.youtube.com/watch?v=WZemrM46AGU>, 22 d'octubre del 2015]

[17] **Microsoft. Random.Next (Método)** [[https://msdn.microsoft.com/es-es/library/system.random.next\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.random.next(v=vs.110).aspx), 3 de novembre del 2015]